



API Document

Cybersecurity Recommendation

Mandatory actions to be taken towards cybersecurity

1. Change Passwords and Use Strong Passwords:

The number one reason systems get "hacked" is due to having weak or default passwords. It is recommended to change default passwords immediately and choose a strong password whenever possible. A strong password should be made up of at least 8 characters and a combination of special characters, numbers, and upper and lower case letters.

2. Update Firmware

As is standard procedure in the tech-industry, we recommend keeping NVR, DVR, and IP camera firmware up-to-date to ensure the system is current with the latest security patches and fixes.

"Nice to have" recommendations to improve your network security

1. Change Passwords Regularly

Regularly change the credentials to your devices to help ensure that only authorized users are able to access the system.

2. Change Default HTTP and TCP Ports:

- Change default HTTP and TCP ports for systems. These are the two ports used to communicate and to view video feeds remotely.
- These ports can be changed to any set of numbers between 1025-65535. Avoiding the default ports reduces the risk of outsiders being able to guess which ports you are using.

3. Enable HTTPS/SSL:

Set up an SSL Certificate to enable HTTPS. This will encrypt all communication between your devices and recorder.

4. Enable IP Filter:

Enabling your IP filter will prevent everyone, except those with specified IP addresses, from accessing the system.

5. Change ONVIF Password:

On older IP Camera firmware, the ONVIF password does not change when you change the system's credentials. You will need to either update the camera's firmware to the latest revision or manually change the ONVIF password.

6. Forward Only Ports You Need:

- Only forward the HTTP and TCP ports that you need to use. Do not forward a huge range of numbers to the device. Do not DMZ the device's IP address.
- You do not need to forward any ports for individual cameras if they are all connected to a recorder on site; just the NVR is needed.

7. Limit Features of Guest Accounts:

If your system is set up for multiple users, ensure that each user only has rights to features and functions they need to use to perform their job.

8. UPnP:

- UPnP will automatically try to forward ports in your router or modem. Normally this would be a good thing. However, if your system automatically forwards the ports and you leave the credentials defaulted, you may end up with unwanted visitors.
- If you manually forwarded the HTTP and TCP ports in your router/modem, this feature should be turned off regardless. Disabling UPnP is recommended when the function is not used in real applications.

9. SNMP:

Disable SNMP if you are not using it. If you are using SNMP, you should do so only temporarily, for tracing and testing purposes only.

10. Multicast:

Multicast is used to share video streams between two recorders. Currently there are no known issues involving Multicast, but if you are not using this feature, deactivation can enhance your network security.

11. Check the Log:

If you think that someone has gained unauthorized access to your system, you can check the system log. The system log will show you which IP addresses were used to login to your system and what was accessed.

12. Physically Lock Down the Device:

Ideally, you want to prevent any unauthorized physical access to your system. The best way to achieve this is to install the recorder in a lockbox, locking server rack, or in a room that is behind a lock and key.

13. Connect IP Cameras to the PoE Ports on the Back of an NVR:

Cameras connected to the PoE ports on the back of an NVR are isolated from the outside world and cannot be accessed directly.

14. Isolate NVR and IP Camera Network

The network your NVR and IP camera resides on should not be the same network as your public computer network. This will prevent any visitors or unwanted guests from getting access to the same network the security system needs in order to function properly.

Foreword

Purpose

Welcome to use API of HTTP protocol specification.

Reader

- API software development engineers
- Project managers
- Product managers

Safety Instructions

The following categorized signal words with defined meaning might appear in the Manual.

Signal Words	Meaning
 DANGER	Indicates a high potential hazard which, if not avoided, will result in death or serious injury.
 WARNING	Indicates a medium or low potential hazard which, if not avoided, could result in slight or moderate injury.
 CAUTION	Indicates a potential risk which, if not avoided, could result in property damage, data loss, lower performance, or unpredictable result.
 TIPS	Provides methods to help you solve a problem or save you time.
 NOTE	Provides additional information as the emphasis and supplement to the text.

Privacy Protection Notice

As the device user or data controller, you might collect personal data of other such as face, fingerprints, car plate number, Email address, phone number, GPS and so on. You need to be in compliance with the local privacy protection laws and regulations to protect the legitimate rights and interests of other people by implementing measures include but not limited to: providing clear and visible identification to inform data subject the existence of surveillance area and providing related contact.

About the Manual

- The Manual is for reference only. If there is inconsistency between the Manual and the actual product, the actual product shall prevail.
- We are not liable for any loss caused by the operations that do not comply with the Manual.
- The Manual would be updated according to the latest laws and regulations of related regions. For detailed information, see the paper manual, CD-ROM, QR code or our official website. If

there is inconsistency between paper manual and the electronic version, the electronic version shall prevail.

- All the designs and software are subject to change without prior written notice. The product updates might cause some differences between the actual product and the Manual. Please contact the customer service for the latest program and supplementary documentation.
- There still might be deviation in technical data, functions and operations description, or errors in print. If there is any doubt or dispute, please refer to our final explanation.
- Upgrade the reader software or try other mainstream reader software if the Manual (in PDF format) cannot be opened.
- All trademarks, registered trademarks and the company names in the Manual are the properties of their respective owners.
- Please visit our website, contact the supplier or customer service if there is any problem occurred when using the device.
- If there is any uncertainty or controversy, please refer to our final explanation.

Table of Contents

Cybersecurity Recommendations.....	I
Foreword	III
Table of Contents	VIII
1 Overview	23
2 References.....	24
3 Definitions.....	25
3.1 Abbreviations	25
3.2 Syntax Convention.....	25
3.3 HTTP API Protocol Format.....	25
3.3.1 key=value format.....	26
3.3.2 JSON format.....	27
3.4 Authentication.....	30
3.5 Conventions.....	31
3.5.1 Channels.....	31
4 General APIs.....	32
4.1 APIs of RTSP.....	32
4.1.1 Get Real-Time Stream.....	32
4.1.2 Get Playback Stream	34
4.1.3 Get File Stream.....	34
4.1.4 Get MJPG Stream.....	34
4.2 Configure Manager.....	35
4.2.1 Get and Set Configure	36
4.2.2 Add and Remove Configure.....	37
4.2.3 Restore the Config	38
4.2.4 Restore except the Config.....	38
4.3 Audio.....	39
4.3.1 Get Audio Input Channel Numbers	39
4.3.2 Get Audio Output Channel Numbers	39
4.3.3 Post Audio Stream.....	39
4.3.4 Get Audio Stream.....	41
4.3.5 [Config] Volume Control.....	42
4.4 Snapshot.....	42
4.4.1 [Config] Snap.....	66
4.4.2 Get a Snapshot.....	68
4.4.3 Subscribe to Snapshot.....	68
4.5 Video Attributes	71
4.5.1 Get Max Extra Stream Numbers	71
4.5.2 Get Encode Capability	71
4.5.3 Get Encode Config Capability.....	72
4.5.4 [Config] Encode of Media	76
4.5.5 [Config] Encode of Region Interested.....	81
4.5.6 [Config] Channel Title.....	82

4.5.7 Get Video Input Channels Device Supported.....	83
4.5.8 Get Video Output Channels Device Supported.....	83
4.5.9 Get Max Remote Input Channels.....	83
4.5.10 [Config] Video Standard	83
4.5.11 [Config] Video Widget.....	84
4.5.12 Get Video Input Capability.....	87
4.5.13 Get Coordinates of Current Window.....	90
4.5.14 Set Coordinates of Current Window	91
4.5.15 [Config] Video Out.....	91
4.5.16 [Config] Smart Encode.....	92
4.5.17 Get Decoder Caps.....	93
4.5.18 [Config] PrivacyMasking.....	94
4.5.19 Get Privacy Masking.....	96
4.5.20 Set Privacy Masking	97
4.5.21 Goto Privacy Masking.....	98
4.5.22 Delete Privacy Masking.....	98
4.5.23 Clear Privacy Masking.....	99
4.5.24 Get Privacy Masking Rect.....	99
4.5.25 [Config] Motion Detection Settings.....	99
4.5.26 [Config] LAEConfig.....	105
4.5.27 Enable/Disable all privacy masking covers.....	107
4.5.28 Get enable/disable state of all privacy masking covers	107
4.6 System	108
4.6.1 [Config] General.....	108
4.6.2 Get Current Time.....	109
4.6.3 Set Current Time.....	109
4.6.4 [Config] Locales	109
4.6.5 [Config] Holiday Management.....	111
4.6.6 Get Language Capability	112
4.6.7 [Config] Language	112
4.6.8 Get Device Type	113
4.6.9 Get Hardware Version	113
4.6.10 Get Serial Number of Device.....	114
4.6.11 Get Machine Name.....	114
4.6.12 Get System Information1.....	114
4.6.13 Get System Information2.....	115
4.6.14 Get Vendor Information.....	116
4.6.15 Get Software Information	116
4.6.16 Get Version of Onvif	117
4.6.17 Get Version of HTTP API.....	117
4.6.18 Get Device Class.....	117
4.6.19 [Config] Auto Maintain.....	117
4.6.20 Reboot	119
4.6.21 Shutdown.....	119
4.6.22 Factory Reset.....	119
4.6.23 Get Tracing Code of Device.....	120
4.6.24 Add Camera to Specified Channel.....	120

4.6.25 Acquiring All Available Resources	121
4.6.26 Subscribing for Device Online/Offline Status.....	128
4.7 User management	129
4.7.1 Get Information of a Particular User	133
4.7.2 Get Information of All Users	133
4.7.3 Get Information of All Active Users.....	134
4.7.4 Get Information of a Particular Group.....	134
4.7.5 Get Information of All Groups	134
4.7.6 Add a New User.....	135
4.7.7 Delete a User	135
4.7.8 Modify User Information.....	136
4.7.9 Modify User's Password	136
4.7.10 Manager Modify Common User's Password	136
4.7.11 [Config] Set User Login Authentication Policy	137
4.7.12 Export all the user info	137
4.8 Network	138
4.8.1 Get Network Interfaces	138
4.8.2 Get Client Access Filter	138
4.8.3 [Config] Network Basic Config	139
4.8.4 [Config] PPPoE	140
4.8.5 [Config] DDNS.....	141
4.8.6 [Config] Email.....	144
4.8.7 [Config] WLan.....	145
4.8.8 Scan Wlan Devices	146
4.8.9 [Config] UPnP	147
4.8.10 Get UPnP Status	148
4.8.11 [Config] NTP	148
4.8.12 [Config] RTSP	149
4.8.13 [Config] Alarm Server.....	150
4.8.14 [Config] Onvif Service Authorization	151
4.8.15 [Config] SSHD Config	151
4.8.16 [Config] Cellular Network Traffic Packages.....	152
4.8.17 Obtaining the Traffic Information of Current Month	153
4.8.18 Searching for History Traffic Statistics According to Specified Conditions.....	155
4.9 Event	157
4.9.1 [DataType] Event Handler	157
4.9.2 [Config] Alarm Event.....	161
4.9.3 [Config] Alarm Out	162
4.9.4 Get Alarm Input Channels	163
4.9.5 Get Alarm Output Channels	163
4.9.6 Get States of Alarm Input Channels.....	163
4.9.7 Get States of Alarm Output Channels	163
4.9.8 [Config] Video Blind Event.....	164
4.9.9 [Config] Video Loss Event	165
4.9.10 [Config] Login Failure Event.....	165
4.9.11 [Config] Storage Not Exist Event.....	166
4.9.12 [Config] Storage Access Failure Event.....	167

4.9.13 [Config] Storage Low Space Event.....	168
4.9.14 [Config] Net Abort Event.....	169
4.9.15 [Config] IP Conflict Event.....	169
4.9.16 Get Channels Event Happened.....	170
4.9.17 Subscribe to Event Message.....	171
4.9.18 Get Capability of Event Management.....	173
4.9.19 [Config] Net Alarm Event	174
4.9.20 Set Net Alarm State.....	176
4.9.21 Get Supported Events	176
4.10 Record.....	177
4.10.1 Get Capability of Recording	177
4.10.2 [Config] Record Config	177
4.10.3 [Config] Record Mode.....	178
4.10.4 [Config] Media Global	179
4.10.5 Find Media Files	180
4.10.6 Find media files with FaceDetection info.....	183
4.10.7 Find media files with FaceRecognition info.....	186
4.10.8 Find media files with HumanTrait info.....	189
4.10.9 Find media files with TrafficCar info.....	193
4.10.10 Find media files with IVS info	196
4.10.11 Find media files with NonMotor info.....	198
4.10.12 Download Media File with the File Name.....	201
4.10.13 Download Media File between Times.....	207
4.10.14 Encrypted Download Media File with the File Name.....	208
4.11 Log	208
4.11.1 Find Logs	209
4.11.2 Clear All the Logs.....	210
4.11.3 Backup Logs.....	211
4.11.4 Seek Find Logs	211
4.11.5 Export Encryped Log.....	212
4.11.6 [Config] Serial Port Log Redirection.....	212
4.12 Upgrader	213
4.12.1 Strat to Upgrade	213
4.12.2 Get Upgrade State.....	214
4.12.3 Set upgrader url.....	214
4.12.4 Cancel Upgrade.....	215
4.12.5 Checking Cloud Update Version	215
4.12.6 Performing Online Update	217
4.12.7 Canceling Online Update	218
4.13 Http Uploading	219
4.13.1 [Config] Active Image and Event Uploading	219
4.13.2 Active Image and Event Uploading	220
4.13.3 [Config] Active Event Uploading	221
4.13.4 Active Event Uploading.....	222
4.13.5 [Config] Active Report Data Uploading	223
4.13.6 People Counting Report Data Uploading	224
4.13.7 Video Structuring Report Data Upload	227

4.13.8 People Flow Heat Map Report Data Upload	228
4.13.9 ANPR Report Data Upload	229
4.13.10 Crowd Distribution Report Data upload	230
4.13.11 Vehicle Density Report Data Upload	231
5 Camera APIs.....	235
5.1 Image.....	235
5.1.1 [Config] Brightness, Contrast and Saturation	235
5.1.2 [Config] Sharpness.....	236
5.1.3 [Config] Flip, Mirror and Rotate90	237
5.2 Exposure.....	238
5.2.1 [Config] Exposure Config.....	238
5.3 Backlight.....	240
5.3.1 [Config] Backlight Config	240
5.4 White Balance ..	241
5.4.1 [Config] White Balance Config.....	241
5.5 Day-Night.....	242
5.5.1 [Config] Day-Night Config.....	242
5.6 Zoom and Focus.....	243
5.6.1 Adjust Focus.....	243
5.6.2 Adjust Focus Continuously.....	244
5.6.3 Auto Focus.....	244
5.6.4 Get Focus Status.....	245
5.6.5 [Config] Zoom Config	245
5.6.6 [Config] Focus Config	246
5.7 Lighting.....	247
5.7.1 [Config] Lighting Config	247
5.8 Video in Options.....	249
5.8.1 Change binocular camera's splice mode.....	249
5.8.2 [Config] Video in Options Config	250
6 Storage APIs.....	259
6.1 Storage Devices.....	259
6.1.1 Get Hard Disk Information.....	259
6.1.2 Get the Name of All Storage Devices	260
6.1.3 Get Storage Device Information	260
6.1.4 Get Storage Capability.....	261
6.1.5 Format Camera SD-Card	261
6.1.6 [Config] Hard Disk Recording Type.....	262
6.2 NAS.....	262
6.2.1 [Config] NAS Information.....	271
6.3 Storage Point.....	272
6.3.1 [Config] Record Storage Point	272
6.3.2 [Config] Storage Group	274
6.4 SDEncrypt.....	275
6.4.1 Encrypt SD Card.....	275
6.4.2 Decrypt SD Card.....	276
6.4.3 Clear SD Card Password	276
6.4.4 Modify SD Card Password	276

6.4.5 Get SD Card Operate Error Policy.....	277
6.4.6 [Config] Storage Health Alarm Settings.....	277
7 Display APIs	279
7.1 GUI.....	279
7.1.1 [Config] GUISet.....	279
7.2 Split Screen	280
7.2.1 Split Screen Mode	280
7.3 Monitor Tour	281
7.3.1 [Config] Monitor Tour.....	281
7.3.2 Enable Tour	282
7.3.3 [Config] Monitor Collection	282
8 Comm APIs.....	284
8.1 PTZ	284
8.1.1 [Config] PTZ Config.....	284
8.1.2 Get PTZ Protocol List.....	285
8.1.3 Get PTZ Capability of Current Protocol.....	285
8.1.4 Get PTZ Status	287
8.1.5 PTZ Control	287
8.1.6 Preset.....	292
8.1.7 Tour	294
8.1.8 Scan	296
8.1.9 Pattern.....	298
8.1.10 Pan	299
8.1.11 [Config] PTZ Auto Movement.....	300
8.1.12 PTZ Restart.....	301
8.1.13 PTZ Reset	301
8.1.14 OSD Menu.....	302
8.1.15 [Config] Set up electronic PTZ.....	303
8.2 Wiper.....	305
8.2.1 Move Continuously.....	306
8.2.2 Stop Move.....	307
8.2.3 Move Once	307
8.3 Illuminator.....	307
8.3.1 [Config] Visible-light Illuminator.....	307
8.4 Flashlight.....	308
8.4.1 [Config] Flashlight Config	308
8.5 Coaxial Control IO	309
8.5.1 Control White Light or Speaker.....	309
8.5.2 Get White Light and Speaker Status	310
8.6 Pir Alarm.....	310
8.6.1 [Config] Pir Parameter	310
8.7 SCADA.....	314
8.7.1 Searching for SCADA Attributes.....	314
8.7.2 Configuring SCADA Attributes.....	316
8.7.3 Obtaining Real-time Data of Monitoring Points.....	318
8.7.4 Configuring Monitoring Points	319
8.7.5 Starting Searching for Historical Data	320

8.7.6 Obtaining Historical Data	321
8.7.7 Stopping Searching for Historical Data	323
8.7.8 Obtaining IDs of External Devices Connected to the Host.....	323
8.8 Gyro	324
8.8.1 get gyroscope info	324
9 Video Analyse APIs.....	326
9.1 Video Analyse Event.....	326
9.1.1 [Event] LeftDetection.....	326
9.1.2 [Event] TakenAwayDetection	326
9.1.3 [Event] WanderDetection.....	327
9.1.4 [Event] StayDetection.....	327
9.1.5 [Event] HumanTrait	328
9.1.6 [Event] CrossLineDetection.....	330
9.1.7 [Event] CrossRegionDetection.....	331
9.1.8 [Event] QueueStayDetection.....	332
9.1.9 [Event] QueueNumDetection	333
9.2 FaceRecognitionServer.....	334
9.2.1 Create Face Group.....	334
9.2.2 Modify Face Group.....	334
9.2.3 Delete Face Group	334
9.2.4 Deploy Face Group	335
9.2.5 Find Face Group.....	337
9.2.6 Re-Abstract Feature By Group	339
9.2.7 Add Person.....	340
9.2.8 Modify Person	340
9.2.9 Delete Person	341
9.2.10 Find Person.....	342
9.2.11 Re-Abstract Features By Person.....	344
9.2.12 [Config] Face Recognition AlarmOut Setting	345
9.2.13 Find Person by Picture	347
9.2.14 Find History Person by Picture	350
9.2.15 [Event] FaceDetection.....	354
9.2.16 [Event] FaceRecognition	355
9.2.17 [Event] FaceFeatureAbstract	357
9.2.18 [Config] Face Recognition Event Handler Setting.....	358
9.2.19 [Config] Face-ID Recognition Threshold	360
9.2.20 Export Face Database	360
9.2.21 Importing Face Database	361
9.3 People Counting.....	362
9.3.1 Get Summary	362
9.3.2 Query the Count of People.....	364
9.3.3 Clear the People Count Information.....	367
9.3.4 Subscribe the People Count Information.....	367
9.3.5 Clear statistics in time section.....	370
9.3.6 [Config] Video Widget Number Status	370
9.3.7 [Event] NumberStat	371
9.3.8 [Event] ManNumDetection.....	372

9.3.9 [Event] CrowdDetection	372
9.4 Heat Map.....	373
9.4.1 Get Heat Map Information.....	373
9.4.2 Get People Heat Map Information.....	374
9.4.3 Subscribe People Realtime Trace Information	375
9.4.4 Get People Histroy Trace Information.....	377
9.4.5 Subscribe Heat Map Raw Data	378
9.5 Crowd Distribute Map.....	379
9.5.1 Get Channel Caps.....	379
9.5.2 Subscribe to Realtime Crowd Stat	379
9.5.3 Get Current Crowd Stat	381
9.6 Video Analyse.....	381
9.6.1 Get Video Analyse Capability.....	381
9.6.2 [Config] Video Analyse Global	382
9.6.3 [Config] Video Analyse Rule.....	384
9.6.4 Get Last Event Info.....	388
9.6.5 [Config] GlobalDeviceParam.....	389
9.6.6 Get Template Rule.....	389
9.6.7 [Config] IntelliSchemeTourEnableSetting	391
9.6.8 [Config] Intelligent Tour Plan	391
9.6.9 Export Intelligent Diagnosis, Allowlist, and Blocklist Information.....	397
9.6.10 Import Intelligent O&M, Allowlist, and Blocklist Information	397
9.6.11 Get Intelligent Capability.....	399
9.6.12 Subscribe Resource Usage Info.....	400
9.6.13 Export Encrypted Files.....	401
9.7 WorkSuitCompareServer	402
9.7.1 Add Compliance Library.....	405
9.7.2 Delete Compliance Library.....	406
9.7.3 Find Compliance Library.....	407
9.7.4 Get Compliance Library Arming Information of Channels.....	408
9.7.5 Modify Compliance Group Information	409
9.7.6 Deploy Compliance Library	410
9.7.7 Find Workwear Information in Compliance Library	410
9.7.8 Get Find Workwear Information Result	411
9.7.9 Stop Find Workwear Information	414
9.7.10 Delete Workwear Information	414
9.7.11 Re-extracting Features by Workwear	415
9.7.12 Stop Re-extracting Workwear Features	416
9.8 Smart Motion Detection	416
9.8.1 [Config] SmartMotionDetect	416
9.8.2 Start SMD Data Search	417
9.8.3 Get SMD Data Search Result	418
9.8.4 End SMD Data Search	419
10 Intelligent Traffic APIs.....	440
10.1 Intelligent Traffic Event.....	440
10.1.1 [Event] TrafficJunction	440
10.1.2 [Event] TrafficRetrograde	442

10.1.3 [Event] TrafficJam.....	442
10.1.4 [Event] TrafficUnderSpeed.....	443
10.1.5 [Event] TrafficOverSpeed.....	443
10.1.6 [Event] TrafficPedestrain	444
10.1.7 [Event] TrafficParking.....	445
10.2 Traffic Flow	445
10.2.1 [Event] TrafficFlowStat.....	445
10.2.2 Find Traffic Flow History	446
10.2.3 Start Traffic Statistics Search.....	447
10.2.4 Get Traffic Statistics	448
10.2.5 End Traffic Statistics Search.....	449
10.3 Traffic Record.....	450
10.3.1 Insert Traffic BlockList/AllowList Record	450
10.3.2 Update Traffic BlockList/AllowList Record.....	451
10.3.3 Remove Traffic BlockList/AllowList Record.....	451
10.3.4 Find Traffic BlockList/AllowList Record.....	452
10.3.5 RemoveEx Traffic BlockList/AllowList Record.....	453
10.3.6 Import Traffic BlockList/AllowList.....	453
10.3.7 Export Traffic BlockList/AllowList.....	454
10.3.8 Export Traffic Flow.....	456
10.3.9 Export Traffic Snap Event Info.....	457
10.4 Traffic Snap Operation.....	459
10.4.1 Open Strobe.....	459
10.4.2 Open/Close Unlicensed Vehicle Detection	459
10.4.3 Manual Snap.....	459
10.5 Traffic Parking	460
10.5.1 Get the Specific Parking Space Status	460
10.5.2 Get All Status of Parking Spaces	461
10.5.3 [Config] Parking Space Light State	461
10.5.4 Set Order State.....	463
10.5.5 Set Light State	463
10.5.6 [Config] Parking Space Access Filter Setting.....	464
10.5.7 Set OverLine State	465
10.6 Vehicles Distribution	465
10.6.1 Subscribe Vehicles Distribution Data.....	465
11 Thermography and Radiometry APIs.....	481
11.1 Thermography Manager.....	481
11.1.1 Get Capability of Thermography.....	481
11.1.2 [Config] Thermography Options.....	482
11.1.3 Get ExternSystem Information.....	484
11.1.4 Get Information of Preset Mode	484
11.1.5 Get Optimized Region Information	485
11.1.6 Enable Shutter	485
11.1.7 Fix Focus	486
11.1.8 Do Flat Field Correction	486
11.2 Radiometry.....	486
11.2.1 Get Capability of Radiometry	486

11.2.2 [Config] Heat Image Thermometry	487
11.2.3 [Config] Thermometry Rule	489
11.2.4 [Config] Heat Image Temper Event.....	491
11.2.5 Get Temperature of Particular Point.....	492
11.2.6 Get Temperature of Particular Condition	493
11.2.7 Find Temperature Information.....	493
11.2.8 Subscribe Temperature Information.....	495
11.2.9 Subscribe Radiometry Data	496
11.2.10 Fetch Radiometry Data.....	498
11.2.11 [Config] Get FireWarning Config.....	498
11.2.12 [Config] Set FireWarning Config.....	499
11.2.13 [Config] Get FireWarningMode Config.....	500
11.2.14 [Config] Set FireWarningMode Config.....	500
11.2.15 Get Current Hot Cold Spot.....	501
11.2.16 [Config] Heat Image Temper PreAlarm Event.....	501
11.2.17 Get Heat Map Info	502
11.3 TemperCustom.....	504
11.3.1 Set Environment Temperature.....	504
12 Access Control APIs	505
12.1 Access Control	505
12.1.1 Open Door.....	505
12.1.2 Close Door.....	505
12.1.3 Get Door Status.....	506
12.1.4 Get Lock Status	506
12.1.5 Capture Fingerprint	506
12.1.6 Capture Face Picture	507
12.1.7 Query AccessControl Record	509
12.1.8 Query Access Control Alarm Record	511
12.1.9 [Event] AccessControl.....	512
12.1.10 [Event] CitizenPictureCompare	515
12.1.11 [Event] Door Status Event.....	517
12.1.12 [Config] Access Control General Setting.....	517
12.1.13 [Config] Access Control Setting.....	520
12.1.14 [Config] Wiegand Setting	523
12.1.15 [Config] Access Time Schedule Setting	525
12.1.16 [Config] Special Day Group Setting	526
12.1.17 [Config] Special Days Schedule Setting	527
12.1.18 [Config] MeasureTemperature Setting.....	528
12.1.19 [Config] CitizenPictureCompare Setting.....	532
12.2 Access Control Manager	533
12.2.1 Get Access Control Capability	533
12.2.2 Add SubController	534
12.2.3 Modify SubController	535
12.2.4 Remove SubController	535
12.2.5 Get SubController Info	535
12.2.6 Get SubController States	536
12.2.7 Set RepeatEnter Route.....	537

12.2.8 Get RepeatEnter Route	537
12.2.9 Set ABlock Route	538
12.2.10 Get ABlock Route.....	539
12.2.11 Get Log Status.....	539
12.2.12 Sync Offline Log	540
12.2.13 Sync SubController Time	540
12.3 Access User Account (V1).....	541
12.3.1 Add Access User Face.....	541
12.3.2 Modify Access User Face.....	541
12.3.3 Delete Access User Face.....	542
12.3.4 Find Access User Face.....	543
12.3.5 Add Access User Card and Fingerprint	544
12.3.6 Modify Access User Card and Fingerprint	547
12.3.7 Delete Access User Card and Fingerprint.....	548
12.3.8 Find Access User Card and Fingerprint	549
12.3.9 Get the Total Number of Records of Access User Card and Fingerprint	553
12.4 Access User Account (V2).....	554
12.4.1 Add Access User	554
12.4.2 Modify Access User.....	555
12.4.3 Delete All Access User.....	557
12.4.4 Delete Multiple Access Users	557
12.4.5 Find Multiple Access Users.....	558
12.4.6 Start Find Access User Related Information.....	559
12.4.7 Get Find Result of Access User Related Information	560
12.4.8 Stop Find Access User Related Information.....	561
12.4.9 Add Multiple Access Cards.....	561
12.4.10 Modify Multiple Access Cards.....	562
12.4.11 Delete All Access Cards.....	563
12.4.12 Delete Multiple Access Cards.....	563
12.4.13 Find Multiple Access Cards	564
12.4.14 Start Find Access User Card Related Information.....	565
12.4.15 Get Find Result of Access User Card Related Information	566
12.4.16 Stop Find Access User Card Related Information	566
12.4.17 Add Multiple Access User Fingerprint.....	566
12.4.18 Modify Access User Fingerprint.....	567
12.4.19 Delete All Access User Fingerprint	568
12.4.20 Delete Multiple Access User Fingerprint	569
12.4.21 Find Access User Fingerprint	569
12.4.22 Add Multiple Access User Face.....	570
12.4.23 Update Multiple Access User Face.....	570
12.4.24 Delete All Access User Face	571
12.4.25 Delete Multiple Access User Face	572
12.4.26 Find Access User Face.....	572
12.4.27 Start Find Access User Face Related Information.....	573
12.4.28 Get Find Result of Access User Face Related Information	573
12.4.29 Stop Find Access User Face Related Information.....	574
12.4.30 Access Control Protocol Capability Query	574

12.5 Admin Password.....	580
12.5.1 Add Access Control Admin Password.....	580
12.5.2 Modify Access Control Admin Password.....	580
12.5.3 Delete Access Control Admin Password.....	581
12.5.4 Find Access Control Admin Password.....	582
12.5.5 Get the Total Number of Records of Access Control Admin Password	584
13 Intelligent Building APIs.....	585
13.1 Video Talk.....	585
13.1.1 Subscribe Video Talk Status	585
13.1.2 Unsubscribe Video Talk Status	587
13.1.3 Invite Server on Video Talk	587
13.1.4 Cancel the Video Talk.....	589
13.1.5 Answer the Invitation.....	589
13.1.6 Refuse to Answer the Video Talk Invitation.....	590
13.1.7 Hang Up.....	591
13.2 Intelligent Building Record.....	591
13.2.1 Query Video Talk Log Record.....	591
13.2.2 Insert Announcement Record	592
13.2.3 Query Alarm Record	592
13.3 SIP	593
13.3.1 [Config] SIP Configuration.....	593
13.3.2 [Config] Registrar Configuration	597
13.4 Room Number Database Management.....	599
13.4.1 Adding Room Number	599
13.4.2 Getting Records by Video Talk short number.....	600
13.4.3 Getting Records by reINDo.....	602
13.4.4 Updating Room Number Records.....	604
13.4.5 Deleting Records by reINDo	606
13.4.6 Clearing All Room Numbers.....	606
13.4.7 Getting Total Quantity of Room Number	606
13.5 ElevatorFloorCounter	607
13.5.1 Set Elevator Floor Info	607
13.5.2 Get Elevator WorkInfo	607
13.5.3 Get Capability	608
14 DVR Custom APIs.....	609
14.1 File Finder.....	609
14.1.1 Create a File Finder	609
14.1.2 Create a Motion File Finder	610
14.1.3 Get the File Information Found by the Finder.....	611
14.1.4 Stop the Finder	612
14.1.5 Get Bound Files.....	612
14.2 BandLimit.....	613
14.2.1 Get Bandwidth Limit State.....	613
14.3 Record Files Protection.....	613
14.3.1 Add Protection.....	613
14.3.2 Cancel Protection	614
14.3.3 Remove Protection.....	614

14.3.4 DownloadFile	614
14.3.5 UploadFile	615
14.3.6 List all elements in the specified directory	616
14.4 Daylight	617
14.4.1 Get Daylight.....	617
15 Other APIs.....	618
15.1 Discover Devices	618
15.1.1 Discover Devices on Internet.....	618
15.2 Open Platform	619
15.2.1 Application Start and Stop.....	619
15.2.2 Install Application.....	619
15.2.3 Update Application and License	620
15.2.4 Uninstall Application	622
15.2.5 Download Application Log.....	622
15.3 GPS	622
15.3.1 Get Capablitly	622
15.3.2 [Config] GPS config.....	623
15.3.3 Get GPS Status	623
15.4 Lens Function.....	624
15.4.1 Get Lens Capability.....	624
15.4.2 Adjust Angle Continuously.....	624
15.4.3 Stop Adjusting Angle	625
15.4.4 Adjust Depth Field	625
15.4.5 Adjust Depth Field Continuously	625
15.4.6 Get Depth Field Status	626
15.4.7 Auto Adjust Depth Field	626
15.4.8 Scene Correction.....	627
15.4.9 Reset Angle.....	628
15.5 FishEye	628
15.5.1 Get FishEye Capability	628
15.5.2 [Config] FishEye Setting.....	629
15.6 Radar Adaptor.....	630
15.6.1 Get Radar Capability.....	630
15.6.2 Get Radar Capbility (Enhanced)	630
15.6.3 Get Status.....	631
15.6.4 Calculate Real Size.....	631
15.6.5 Subscribe Alarm Point Info.....	631
15.6.6 Manual Locate	633
15.6.7 Start Radar Calibration	633
15.6.8 Add Radar Link SD.....	633
15.6.9 Del Radar Link SD.....	634
15.6.10 Get Link SD State.....	635
15.6.11 [Config] MapPara.....	635
15.6.12 [Config] RadarAnalyseRule.....	637
15.6.13 [Config] RadarCalibration.....	638
15.6.14 [Config] RadarGuardLine	639
15.6.15 [Config] RadarLink	640

15.6.16 [Config] RadarLinkDevice	640
15.6.17 [Config] RadarPara	640
15.6.18 [Config] RadarTrackGlobal.....	641
15.6.19 [Config] RemoteSDLink.....	642
15.7 Water Radar	642
15.7.1 Acquire Radar Capability.....	642
15.7.2 Get Radar Detection Target Data.....	642
15.8 Water Quality Detection	643
15.8.1 Get Capability	643
15.8.2 Get Real-time Detection Data.....	643
15.8.3 Start Find Water Quality Report	645
15.8.4 Get Find Water QUality Report Search Results.....	646
15.8.5 Stop Find Water Quality Report.....	648
15.9 Advertisement.....	648
15.9.1 Changing Stay Time of Advertisement Image	648
15.9.2 Acquiring Video and Image Files Played by the Device.....	649

Overview

This document specifies the HTTP-based application programming interface of video products.

The HTTP-based interface provides the functionality for requesting snapshot and media stream, controlling camera functions (for example, PTZ and focus), and getting and setting internal parameter values.

The video products serve as a server. The client sends requests to server, and then server handles requests and returns resources accordingly.

2. References

- [1] RFC 2326 Real Time Streaming Protocol (RTSP)
- [2] RFC 2616 Hypertext Transfer Protocol-HTTP/1.1
- [3] RFC 3986 Uniform Resource Identifiers (URI) Generic Syntax
- [4] RFC 7616 HTTP Digest Access Authentication
- [5] RFC 8259 The JavaScript Object Notation (JSON) Data Interchange Format

3.Definitions

3.1 Abbreviations

The following abbreviations are used throughout this document.

API	Application Programming Interface. In the document, it especially presents application programming interface of video products.
-----	---

3.2 Syntax Convention

- In URL syntax and in descriptions of API parameters, text in italic within angle brackets denotes content that should be replaced with either a value or a string. When replacing the text string, the angle brackets must also be replaced. For example, <server> in URL syntax is replaced with the IP number of server, e.g., 192.168.1.250.
- String shown in bold face denotes a brief explanatory note of the string close to it.
- Name-value pair in square brackets denotes content that is optional. For example, "http ://<server>/cgi-bin/snapshot.cgi[?channel=1]" can be like this "http ://<server>/cgi-bin/snapshot.cgi".
- The API syntax must follow the standard of URI. (RFC 3986: Uniform Resource Identifiers (URI) Generic Syntax); that is, spaces and other reserved characters (e.g, ":" , "/" , "?" , "@" , ";" , "=" , "+" , "," , "\$" , "&") within a name-value pair should be replaced with %< ASCII hex>. For example, the blank should be replaced with %20.
- To describe the range of a variable, we use some symbols such as "[]" and "{}". For example, "[0-100]" denotes an integer not less than 0 and not larger than 100. "{0, 1, 2, 3}" denotes the valid value of an integer among 0, 1, 2 and 3.
- "[]" following a string denotes an array. The index is an integer and starts from 0. For example, "Snap[channel]" may be "Snap[0]" "Snap[0]" etc.
- The variable may be different types: string, integer, bool or float. Integer is 32 bits. The range of bool is "true" and "false".
- "R/O" in parameters means this parameter is required or not, "R"means required, "O"means optional.

3.3 HTTP API Protocol Format

HTTP APIs support HTTP and HTTPS protocol. It is recommend using HTTPS to improve network security.

The request and response parameters of the HTTP APIs have two formats: key=value format and JSON format. Each HTTP API will choose one of these formats, please refer to the API detail in later chapters.

3.3.1 key=value format

If the HTTP API use key=value format for it's parameters, the URL syntax is :

```
<protocol>://<server><abs_path>[?query]
```

protocol: URL scheme for the particular request. The http protocol and https protocol are both supported in this specification. So "http", as most of the APIs' default protocol except several RTSP APIs, can be replaced by "https".

server: Server could be "**hostname[:port]**". The **hostname** can be IP address or the fully qualified domain name of an IP device. The **port** is the port number of **server** listening for TCP connections. If the port is not given, the default port is assumed. For HTTP, the default port is 80. For HTTPS, the default port is 443.

abs_path: The Request-URI for the resources is abs_path. The abs_path in this specification is most often of the form "/cgi-bin/*.cgi".

query: The query field is a string of information to be interpreted by the resource. It consists of resource-related parameters. And it must be listed in name-value pair syntax (p1=v1&p2=v2&...&pn=vn).

For the HTTP API that use key=value format for it's parameters, it usually has no HTTP Body.

Request Example: HTTP request with key=value format parameters

```
GET http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=getFocusStatus&channel=1 HTTP/1.1
User-Agent: client/1.0
Content-Length: 0
```

The server uses the standard HTTP status codes, the syntax of the status line of the response is:

```
HTTP/1.1 <HTTP code> <HTTP text>
```

With the following HTTP code and meanings:

HTTP code	HTTP text	Description
200	OK	The request has succeeded. The requested resource will be returned in the HTTP text.
400	Bad Request	The request had bad syntax or was inherently impossible to be satisfied.
401	Unauthorized	The request requires user authentication or the authorization has been refused.
403	Forbidden	The user does not have the right to access the service.
404	Not Found	The server has not found anything matching the request.
500	Internal Server Error	The server encountered an unexpected condition that prevented it from fulfilling the request.
501	Not Implemented	The server has not implemented the service.

If the HTTP code is 200, means the API execute success, and the response data in HTTP Body (maybe multipart) can be a multiline key=value data, or just a line with a word "OK".

Reponse Example: success response with multiline key=value

```
HTTP/1.1 200 OK
Server: Device/1.0
Content-Type: text/plain
Content-Length: <length>
```

```
status.Focus=0.5
status.Zoom=0.5
...
```

Reponse Example: success response with a word "OK"

```
HTTP/1.1 200 OK
Server: Device/1.0
Content-Type: text/plain
Content-Length: <length>
```

```
OK
```

If the HTTP code is not 200, means the API execute failed, and the response data in HTTP Body maybe empty, or just two line, first line is a word "Error" to indicate error happened, the second line contain error detail.

Reponse Example: request does not fit with syntax.

```
HTTP/1.1 404 Not Found
Server: Device/1.0
```

Reponse Example: Request spells wrong.

```
HTTP/1.1 400 Bad Request
Server: Device/1.0
Content-Type: text/plain
Content-Length: <length>
```

```
Error
Bad Request!
```

Reponse Example: If the request fits with syntax but an error occurs while the server handles it, the response would like this:

```
HTTP/1.1 500 Internal Server Error
Server: Device/1.0
Content-Type: text/plain
Content-Length: <length>
```

```
Error
Internal Server Error!
```

3.3.2 JSON format

If the HTTP API use JSON format for its parameters, the URL syntax is :

```
<protocol>://<server><abs_path>
```

protocol: URL scheme for the particular request. The http and https protocols are both supported in this specification. So "http", as most of the APIs' default protocol except several RTSP APIs, can be replaced by "https".

server: Server could be "**hostname[:port]**". The **hostname** can be IP address or the fully qualified domain name of an IP device. The **port** is the port number of **server** listening for TCP connections. If the port is not given, the default port is assumed. For HTTP, the default port is 80. For HTTPS, the default port is 443.

abs_path: The Request-URI for the resources is abs_path. The abs_path in this specification is most often prefix with "/cgi-bin/api/" , and followed with resource name, for example : "/cgi-bin/api/CloudUpgrader/check" .

For the HTTP API that use JSON format for it's parameters, the parameters are put to the HTTP Body as a JSON object. If the API has no request parameter, then the HTTP Body is an empty JSON object, for example " {} ", or there is no HTTP Body.

Request Example: HTTP request with JSON format parameters

```
GET http://192.168.1.108/cgi-bin/api/CloudUpgrader/check HTTP/1.1
User-Agent: client/1.0
Content-Type: application/json
Content-Length: xxx

{
    "way" : 0,
    "proxy" : {
        "IP" : "10.1.2.3",
        "Port" : 8080
    }
}
```

The server uses the standard HTTP status codes, the syntax of the status line of the response is:

```
HTTP/1.1 <HTTP code> <HTTP text>
```

With the following HTTP code and meanings:

HTTP code	HTTP text	Description
200	OK	The request has succeeded. The requested resource will be returned in the HTTP text.
400	Bad Request	The request had bad syntax or was inherently impossible to be satisfied.
401	Unauthorized	The request requires user authentication or the authorization has been refused.
403	Forbidden	The user does not have the right to access the service.
404	Not Found	The server has not found anything matching the request.
500	Internal Server Error	The server encountered an unexpected condition that prevented it from fulfilling the request.

HTTP code	HTTP text	Description
501	Not Implemented	The server has not implemented the service.

If the HTTP code is 200, means the API execute success, and the response data in HTTP Body (maybe multipart) can be a JSON object. If the API has no response parameter, then the HTTP Body is an empty JSON object, for example " {} ", or there is no HTTP Body.

Reponse Example: success respose with JSON format Body

```
HTTP/1.1 200 OK
Server: Device/1.0
Content-Type: application/json
Content-Length: <length>

{
  "info" : {
    "State" : "None",
    "PackageType" : "all",
    "OldVersion" : "0000",
    "NewVersion" : "0004",
    "Attention" : "What is new",
    "PackageUrl" : "https://example.com/1.zip",
    "Packageld" : "1d2ee7",
    "CheckSum" : "F3D288AB",
    "BuildTime" : "08-10-2018 01:01:02"
  }
}
```

If the HTTP code is not 200, means the API execute failed, and the response data in HTTP Body maybe empty, or a JSON object to describe the error detail. The value of "ErrorCode" contains error code, and the value of "ErrorMsg" contains error detail.

Reponse Example: request does not fit with syntax.

```
HTTP/1.1 404 Not Found
Server: Device/1.0
```

Reponse Example: Request spells wrong.

```
HTTP/1.1 400 Bad Request
Server: Device/1.0
Content-Type: application/json
Content-Length: <length>

{
  "ErrorCode" : 10086,
  "ErrorMsg" : "Bad Request!"
}
```

Reponse Example: If the request fits with syntax but an error occurs while the server handles it, the response would like this:

```
HTTP/1.1 500 Internal Server Error
```

```
Server: Device/1.0
```

```
Content-Type: application/json
```

```
Content-Length: <length>
```

```
{
```

```
    "ErrorCode" : 10097
```

```
    "ErrorMsg" : "Internal Server Error!"
```

```
}
```

3.4 Authentication

Video products support digest authentication, see RFC 7616 for detail. If the http request sent by client does not provide valid "Authorization" header information, video products would return HTTP status code 401 and some information for authentication, then client should calculate authentication information according to RFC 7616, and sent request again with authentication information using "Authorization" header. Video products return the required resource only if authorization information correct.

1. When the client does not provoding digest authentication or the client calculates digest authentication with expired "nonce" data, video product reply 401 with information for authorization.

Request Example: The client send request without authentication infomation

```
GET /cgi-bin/magicBox.cgi?action=getLanguageCaps HTTP/1.1
```

```
User-Agent: client/1.0
```

```
Content-Length: 0
```

Reponse Example: The device send 401 reponse with parameters for authentication calculation.

```
HTTP/1.1 401 Unauthorized
```

```
Server: Device/1.0
```

```
WWW-Authenticate: Digest realm="Device_00408CA5EA04",
```

```
    nonce="000562fd20ef95ad", qop="auth",
```

```
    opaque="5ccc069c403ebaf9f0171e9517f40e41"
```

2. Then the client generate "nc" and "INDonce", calculates the digest authorization using "username" "password" "HTTP method" "URI" "nc" "INDonce" "realm" "nonce" , according to RFC 7616, and then sends it to video product again. If the digest authorization is correct, the video product will reply HTTP status code 200 with the response data. If digest authorization is not correct, the video product will reply HTTP status code 403 .

Request Example: The client send request again with authentication information (suppose the password of admin user is "abcd1234")

```
GET /cgi-bin/magicBox.cgi?action=getLanguageCaps HTTP/1.1
User-Agent: client/1.0
Authorization: Digest username="admin", realm="Device_00408CA5EA04",
nonce="000562fd20ef95ad", nc=00000001, cnonce="0a4f113b", qop="auth",
uri="/cgi-bin/magicBox.cgi?action=getLanguageCaps",
response="dfd0f24bed4c336d20c8f0729dd5dbc8"
opaque="5ccc069c403ebaf9f0171e9517f40e41"
Content-Length: 0
```

Reponse Example: If the authentication information is correct, the device send 200 reponse.

```
HTTP/1.1 200 OK
Server: Device/1.0
Content-Type: text/plain
Content-Length: <length>

Languages=SimpChinese,English,French
```

Reponse Example: If the authentication information is not correct, the device send 403 reponse.

```
HTTP/1.1 403 Forbidden
Server: Device/1.0
Content-Length: 0
```

3.5 Conventions

3.5.1 Channels

For requests containing the param 'channel' like :

`http://<server>/cgi-bin/mjpg/video.cgi[?channel=<ChannelNo>]`

ChannelNo: integer, must starts from 1. Default is 1 if not present.

But for responses, the '**ChannelNo**' should start from 0. In other words, the request channel 1 equals response channel 0.

4.General APIs

The APIs specified in this section are supported by all video products.

4.1 APIs of RTSP

4.1.1 Get Real-Time Stream

Get real-time media stream APIs use RTSP protocol, please refer to RFC 2326 for detail. The rtsp service default port is 554. The IP Camera supports both TCP and UDP transmission forms. It supplies digest authentication ways. The authentication process is similar with "3.5 Authentication".

- The format of the RTSP URL parameter for getting real-time media stream is as follows.

URL	rtsp://<server>:[port]/cam/realmonitor			
Method	DESCRIBE, SETUP, PLAY, PAUSE, TEARDOWN, ...			
URL Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	Video channel number, starting from 1.	1
subtype	int	R	<p>The stream type, including main stream and sub stream.</p> <p>Execute the "4.5.1 Getting Maximum Number of Sub Streams" command to get the number of sub streams.</p> <p>Optional values:</p> <p>0: Main stream 1: Sub stream 1 2: Sub stream 2</p>	0

URL Example

rtsp://192.168.1.250:554/cam/realmonitor?channel=1&subtype=0

Step 1: Obtain media description by executing the DESCRIBE command.

Request Example

DESCRIBE rtsp://192.168.1.250/cam/realmonitor?channel=1&subtype=0 RTSP/1.0
CSeq: 1

User-Agent: LibVLC/3.0.5

Response Example

RTSP/1.0 200 OK
CSeq: 1
Server: Rtsp Server/3.0
Content-Base: rtsp://192.168.1.250/cam/realmonitor?channel=1&subtype=0/
Content-Length: xxx
Content-Type: application/sdp

```
v=0
o=- 2253484289 2253484289 IN IP4 0.0.0.0
s=Media Server
c=IN IP4 0.0.0.0
t=0 0
a=control:*
a=range:npt=now-
m=video 0 RTP/AVP 98
a=control:trackID=0
a=framerate:25.000000
a=rtpmap:98 H265/90000
a=recvonly
```

Step 2: Establish a transmission channel for each media by executing the SETUP command.

Take RTP over UDP as an example. Establish the UDP socket for receiving and sending UDP packages on 63088 and 63089 interfaces.

Request Example

```
SETUP rtsp://192.168.1.250/cam/realmonitor?channel=1&subtype=0/trackID=0 RTSP/1.0
CSeq: 2
User-Agent: LibVLC/3.0.5
Transport: RTP/AVP;unicast;client_port=63088-63089
```

Response Example

```
RTSP/1.0 200 OK
CSeq: 2
Server: Rtsp Server/3.0
Session: 1546116282447;timeout=60
Transport: RTP/AVP/UDP;unicast;client_port=63088-63089;server_port=24764-24765:ssrc=71B0AFDC
```

Step 3: Execute the play command to play the media, and receive and send RTP and RTCP data through the UDP socket established in step 2.

Request Example

```
PLAY rtsp://192.168.1.250/cam/realmonitor?channel=1&subtype=0/ RTSP/1.0
CSeq: 3
User-Agent: LibVLC/3.0.5
Session: 1546116282447
Range: npt=0.000-
```

Response Example

```
RTSP/1.0 200 OK
CSeq: 3
Server: Rtsp Server/3.0
Session: 1546116282447
Range: npt=0.000-
RTP-Info: url=trackID=0;seq=45020;rtpTime=1907404764
```

Step 4: Execute the TEARDOWN command to stop playing the media, and then disable the UDP socket.

Request Example

```
TEARDOWN rtsp://192.168.1.250/cam/realmonitor?channel=1&subtype=0/ RTSP/1.0
CSeq: 4
User-Agent: LibVLC/3.0.5
Session: 1546116282447
```

Response Example

```
RTSP/1.0 200 OK
CSeq: 4
Server: Rtsp Server/3.0
Session: 1546116282447
```

4.1.2 Get Playback Stream

It's similar with "4.1.1 Get Real-Time Stream", except that there are parameters "starttime" and "endtime".

URL	rtsp://<server>:[port]/cam/playback			
Method	DESCRIBE, SETUP, PLAY, PAUSE, TEARDOWN, ...			
URL Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	The video channel index which starts from 1	1
starttime	char[32]	R	The playback begin time	2012_09_15_12_37_05
endtime	char[32]	O	The playback end time	2012_09_15_18_34_14
URL Example				rtsp://192.168.1.250:554/cam/playback?channel=1&starttime=2012_09_15_12_37_05&endtime=2012_09_15_18_34_14

4.1.3 Get File Stream

It's similar with "4.1.1 Get Real-Time Stream". The filename with absolute path of file stream to get is place in rtsp url.

URL	rtsp://<server>:[port]/<path to filename>			
Method	DESCRIBE, SETUP, PLAY, PAUSE, TEARDOWN, ...			
URL Params (key=value format in URL)				
Name	Type	R/O	Description	Example
URL Example				rtsp://192.168.1.250:554//mnt/sd/2015-09-16/001/dav/20/20.32.08-20.32.28[M][0@0][0].dav

4.1.4 Get MJPG Stream

Get a video stream encoded by mjpg (motion jpeg).

Request URL	http://<server>/cgi-bin/mjpg/video.cgi
--------------------	--

Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	The video channel index which starts from 1	1
subtype	int	R	The stream type, It can be the following value: 0-Main Stream 1-Extra Stream 1 2-Extra Stream 2	0
Request Example				
http://192.168.1.250/cgi-bin/mjpg/video.cgi?channel=1&subtype=0				

Response Params (multipart, binary in body)
<binary data> : JPEG image data
Response Example
HTTP/1.1 200 OK
Server: Device/1.0
Content-Type: multipart/x-mixed-replace; boundary=<boundary>
--<boundary>
Content-Type: image/jpeg
Content-Length:<image size>
<JPEG image data>
--<boundary>
Content-Type: image/jpeg
Content-Length:<image size>
<JPEG image data>
--<boundary>
.....

4.1.5 Create RTSP Pull and Authentication Token

Request URL	http://<server>/cgi-bin/api TokenNameManager/createToken			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Channel	uint32	R	video channel number which starts from 0	0
Request Example				
{				
"Channel": 0				
}				
Response Params (key=value format in body)				

Name	Type	R/O	Description	Example
Token	char[128]	R	Token	"AwDSEci0j4EUW6pUz5bcQ0yK8_Rbq0vaUsUwqQKmFoUpuMjOxy9kgcV6BICty8U"
Response Example				
{				"Token": "AwDSEci0j4EUW6pUz5bcQ0yK8_Rbq0vaUsUwqQKmFoUpuMjOxy9kgcV6BICty8U"

4.2 Configure Manager

4.2.1 Get and Set Configure

- Get Configure

Get configure detail by name. Each configure name has it's specific configure detail. Please refer to configure detail APIs in later [Config] chapters.

Request URL	http://<server>/cgi-bin/configManager.cgi?action=getConfig		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
name	string	R	The name of the configure
Request Example			
http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=General			

Response Params (key=value format in body)				
Name	Type	R/O	Description	Sample
table	object	R	The configure table object.	
Response Example				
+<config name>	string/object/object[]	R	The configure detail of the requested configure name. Each configure name has it's specific configure detail. Please refer to configure detail apis in later chapters.	
table.General.MachineID=20832748927				
table.General.MachineName=DVR001				
table.General.MachineAddress=XXX Road				
...				

- Set Configure

Set configure detail by name. Each configure name has it's specific configure detail. Please refer to configure detail APIs in later [Config] chapters.

Request URL	http://<server>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Request Params (key=value format in URL)			

Name	Type	R/O	Description	Sample
<config name>	string/object/object[]	R	The configure detail of the configure name. Each configure name has it's specific configure detail. Please refer to configure detail apis in later chapters.	

Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&General.MachineID=20832748927&General.MachineName=DVR001&General.MachineAddress=XXX%20Road
```

Response Params (OK in body)

Response Example

```
OK
```

4.2.2 Add and Remove Configure

- Add configure

The configuration name is put in the URL in the form of key-value; the specific configuration field can be obtained through getConfig. Each configure name has it's specific configure detail. Please refer to configure detail APIs in later [Config] chapters.

Request URL	http://<server>/cgi-bin/configManager.cgi?action=addConfig			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample
<config name>	string/object/object[]	R	The configure detail of the configure name. Each configure name has it's specific configure detail. Please refer to configure detail apis in later chapters.	

Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=addConfig&General.MachineAddress=XXX%20Road
```

Response Params (OK in body)

Response Example

```
OK
```

- Remove configure

Specific configuration fields can be obtained through getConfig. Only configuration fields can be deleted, not configuration items. Each configure name has it's specific configure detail. Please refer to configure detail APIs in later [Config] chapters.

Request URL	http://<server>/cgi-bin/configManager.cgi?action=removeConfig			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample
name	string	R	configuration param name	VSP_CGI.ServiceStart

Request Example

http://192.168.1.250/cgi-bin/configManager.cgi?action=removeConfig&General.MachineAddress

Response Params (OK in body)

Response Example

OK

4.2.3 Restore the Config

Restore specified configurations to the default values.

Request URL	http://<server>/cgi-bin/configManager.cgi?action=restore		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
names	char[][32]	R	Array of the configuration names that needs to be restored to the default values.

Request Example

http://<server>/cgi-bin/configManager.cgi?action=restore&names[0]=General&names[1]=UPnp

Response Params (OK in body)

Response Example

OK

4.2.4 Restore except the Config

Restore all other configurations except for the specified configurations to the default values.

Request URL	http://<server>/cgi-bin/configManager.cgi?action=restoreExcept		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
names	char[][32]	R	Array of the configuration names that does not need to be restored to the default values.

Request Example

http://<server>/cgi-bin/configManager.cgi?action=restoreExcept&names[0]=General&names[1]=UPnp

Response Params (OK in body)

Response Example

OK

4.3 Audio

4.3.1 Get Audio Input Channel Numbers

Get audio input channel number.

Request URL	http://<server>/cgi-bin/devAudioInput.cgi?action=getCollect		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.250/cgi-bin/devAudioInput.cgi?action=getCollect			

Response Params (key=value format in body)			
Name	Type	R/O	Description
result	int	R	audio input channel number.
Response Example			
result=1			

4.3.2 Get Audio Output Channel Numbers

Get audio output channel number.

Request URL	http://<server>/cgi-bin/devAudioOutput.cgi?action=getCollect		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.250/cgi-bin/devAudioOutput.cgi?action=getCollect			

Response Params (key=value format in body)			
Name	Type	R/O	Description
result	int	R	audio output channel number.
Response Example			
result=2			

4.3.3 Post Audio Stream

Post audio stream to device, request is very long and client continues to send audio data. If client want to stop, just close the connection.

Request URL	http://<server>/cgi-bin/audio.cgi?action=postAudio		
Method	POST		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
httptype	string	R	audio http transmit format, can be : singlepart: HTTP content is a continuous flow of

			audio packets multipart: HTTP content type is multipart/x-mixed-replace, and each audio packet ends with a boundary string	
channel	int	R	The audio channel index which starts from 1.	1

Request Example (singlepart)

POST http://192.168.1.250/cgi-bin/audio.cgi?action=postAudio&httpType=singlepart&channel=1 HTTP/1.1

User-Agent: client/1.0

Content-Type: Audio/G.711A

Content-Length: 9999999

<Audio data>

<Audio data>

...

Request Example (multipart)

POST http://192.168.1.250/cgi-bin/audio.cgi?action=postAudio&httpType=multipart&channel=1 HTTP/1.1

User-Agent: client/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

--<boundary>

Content-Type: Audio/G.711A

Content-Length: 800

<Audio data>

--<boundary>

Content-Type: Audio/G.711A

Content-Length: 800

<Audio data>

--<boundary>

...

Response Params (N/A)

Name	Type	R/O	Description	Example
Response Example (N/A)				

Appendix A: Audio Encode Type

MIME (Content-Type)	Description
Audio/PCM	PCM
Audio/ADPCM	ADPCM
Audio/G.711A	G.711 A Law
Audio/G.711Mu	G.711 Mu Law
Audio/G.726	G.726

MIME (Content-Type)	Description
Audio/G.729	G.729
Audio/MPEG2	MPEG2
Audio/AMR	AMR
Audio/AAC	AAC

4.3.4 Get Audio Stream

Get audio stream from device, response is very long and client continues to recv audio data. If client want to stop, just close the connection.

Request URL	http://<server>/cgi-bin/audio.cgi?action=getAudio		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
httptype	string	R	audio http transmit format, can be : singlepart: HTTP content is a continuous flow of audio packets multipart: HTTP content type is multipart/x-mixed-replace, and each audio packet ends with a boundary string
channel	int	R	The audio channel index which starts from 1.
Request Example			
http://192.168.1.250/cgi-bin/audio.cgi?action=getAudio&httptype=singlepart&channel=1			

Response Params (binary in body)			
Name	Type	R/O	Description
Response Example (singlepart)			
200 OK HTTP/1.1			
Server: device/1.0			
Content-Type: Audio/G.711A			
Content-Length: 9999999			
<Audio data>			
<Audio data>			
...			
Response Example (multipart)			
200 OK HTTP/1.1			
Server: device/1.0			
Content-Type: multipart/x-mixed-replace; boundary=<boundary>			
--<boundary>			
Content-Type: Audio/G.711A			
Content-Length: 800			
<Audio data>			
--<boundary>			

Content-Type: Audio/G.711A

Content-Length: 800

<Audio data>

--<boundary>

...

4.3.5 [Config] Volume Control

Volume control configuration parameters:

Config Data Params

Name	Type	R/O	Description	Example
AudioOutputVolume	int[]	R	Audio output volume. Each element of the array represents the volume value of an audio output channel, and the range is [0-100].	[80,50]

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

<http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=AudioOutputVolume>

Get Config Response Example

table.AudioOutputVolume[0]=80

table.AudioOutputVolume[1]=50

...

Set Config Request Example

[http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&AudioOutputVolume\[0\]=80&AudioOutputVolume\[1\]=50](http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&AudioOutputVolume[0]=80&AudioOutputVolume[1]=50)

Set Config Response Example

OK

4.3.6 Getting the Analysis Configurations of the Specified Audio Channel

Get the analysis configurations of the specified audio channel.

Request URL	<a href="http://<server>/cgi-bin/api/AudioAnalyseManager/getConfig">http://<server>/cgi-bin/api/AudioAnalyseManager/getConfig			
Method	POST			
Parameter Format	JSON format in body			
Name	Type	Required	Description	Example

PresetID	int32	O	Preset. The field is valid when the analysis channel is specified. When there are no fields, the default preset is 0, and the field can be empty for IPC. When the preset is -1, all the preset configurations will be returned.	0
AudioChannel	int32	O	Audio channel number. When there are no signs indicating that the audio source is from the Internet, then the AudioChannel means the local channel number. When using the number, it must be linked with logical audio channel through channel number or other fields. If it fails to link, the API will be failed to return.	0
DeviceID	char[128]	O	Device ID. It is mainly used in the back-end cascade scenes, and if there are no special requirements, it will not be used in front-end devices.	"12345678"
Complete Example				
{ "PresetID": 0, "AudioChannel": 0, "DeviceID": "12345678" }				

Response Params (JSON format in body)				
Name	Type	Required	Description	Example
Configuration	object	O	Analysis configuration of audio channel	
+AudioChannel	uint32	O	Audio channel number. When there are no signs indicating that the audio	0

			source is from the Internet, then the AudioChannel means the local channel number.	
+Rules	object[]	O	Analysis rule	
++Class	char[64]	O	Rule category	"<ClassType>"
++Type	char[64]	O	Rule type	"<RuleType>"
++Id	uint	O	Rule ID. The field can be 0, which means the system will automatically allocate the appropriate ID. If there are businesses that bonds the rule ID, the client will be responsible for ensuring that the code and the name are unique.	0
++Name	char[64]	O	Rule name	"..."
++Enable	bool	O	Enable the rules	true
++PresetID	uint32	O	Preset spots	0
++EventHandler	object	O	Rule linkage items	
+++TimeSection	char[8][6] [20]	O	Alarm period	[["1 00:00:00-08:00:00", "1 10:00:00-24:00:00",]]
+++RecordEnable	bool	O	Enable recording, with RecordChannels. Record is enabled if it is true; start recording when the event action is start and stop recording when it is stop. Record is disabled if it is false.	true
+++RecordChannels	int[]	O	Recording channel number list One-dimensional array.	[0, 1, 2]

			Each member indicates that the corresponding channel needs to record, and the channel number starts from 0.	
+++RecordLatch	int	O	Recording delay time (second) Range [10, 300]	10
+++AlarmOutputEnable	bool	O	Enable alarm output	true
+++AlarmOutputChannels	int[]	O	Alarm output channel number list One-dimensional array. Each member indicates that the corresponding channel needs to output alarm, and the channel number starts from 0.	[1, 4]
+++AlarmOutputLatch	int	O	Output delay time (second) after the alarm output stops. Range: [1, 300]	10
+++SnapshotEnable	bool	O	Enable snapshot	true
+++SnapshotChannels	int[]	O	Snapshots channel number list One-dimensional array. Each member indicates that the corresponding channel needs to take snapshot, and the channel number starts from 0.	[2, 4]
+++MailEnable	bool	O	Send emails. If there are images, the images will be sent as email attachment.	true
+++Dejitter	int	O	Dejitter time. The field is only meaningful for part of the start or stop type events. Unit: Second (0 s–600 s). It can only be	0

			used for applications to automatically tour the detected events, such as motion detection and tampering. The similar intelligent events that reported by the algorithm database such as tripwire and face detection cannot configure dejitter time.	
+++BeepEnable	bool	O	Buzzer	true
+++Delay	uint	O	Set the delay time for the event to take effect. Unit: second [0-300].	30
+++ExAlarmOutChannels	int[]	O	Expands alarm output channel list One-dimensional array. Each member indicates that the corresponding channel needs to output extension alarm, and the channel number starts from 0.	[2, 3]
+++ExAlarmOutEnable	bool	O	Enable expand alarm output	true
+++LightingLink	object	O	PTZ illuminator linkage item	
++++Enable	bool	O	Enable	true
++++FilckerLightType	enumchar[32]	O	Flicker light type enumchar[32]{ "WhiteLight": indicates the flashing light is white light "RedBlueLight": indicates the flashing light is red and blue light }	"WhiteLight"
++++LightLinkType	enumchar[32]	O	Light linkage way enumchar[32]{ "Filcker": flicker (by	"Filcker"

			default) "KeepLighting": solid on }	
++++FlickerIntervalTime	float	O	Flicker interval Unit: 0.1 s, and the default value is 0.5 s.	5
++++FlickerTimes	int	O	The flicker times that can be configured. Unit: times, and the default value is 5.	5
++++LightDuration	uint	O	The duration time of the flicker or normally lighting. Unit: second.	10
++++LightBrightness	uint	O	Brightness of the linkage light. The range refers to the capacity of LinkLightBrightRange. If this item does not exist, the default brightness will be adopted.	50
+++LogEnable	bool	O	Whether to record alarm logs (Includes local and remote logs). The type is fixed to EventStart/EventStop/EventPulse and the original event type is recorded in the Detail.Code field in the log.	false
+++LogRemote	bool	O	Whether to record remote logs. The type is fixed as EventStart/EventStop/EventPulse and the original event type is recorded in the Detail.Code field in the log. (When LogRemote and LogEnable exist at the same time, the remote log records will use the data from LogRemote).	false

+++MMSEnable	bool	O	Enable sending MMS	false
+++MatrixChannels	int[]	O	Link with video matrix channels number	[1, 6]
+++MatrixEnable	bool	O	Enable linkage video matrix	true
+++MessageEnable	bool	O	Whether to upload the message to the alarm central server. Alarm center does not need to log in to the device, and the device itself can directly send the message to alarm center.	true
+++PtzLink	object[][]	O	PTZ linkage. It is a two-dimensional array, each member indicates a linkage item, and the last item indicates channel number. Each linkage item includes 5 components including linkage type, three relating parameters and the PTZ channel it linked with. Linkage types include: "Preset": linkage preset; "Tour": linkage tour ; "Pattern" : linkage mode; "Zoom" : linkage zoom "SingleScene": link with single scene. (The second linkage parameter indicates the scene number, and the third linkage parameter indicates the stay time of the scene after the alarm finished, the fourth linkage parameter indicates reserve the parameter and the fifth	[["Preset", 3, 0, 0, 0], ["Tour", 4, 0, 0, 1], ["Pattern", 1, 0, 0, 2]]

			<p>linkage parameter indicates the PTZ channel it linked with. "QuickFocus": the thermal imaging PTZ links with quick focus function (the second one indicates the zoom of the thermal imaging channel), and its value ranges from 0 to 24. The third parameter indicates the stay time and the fourth indicates reserve the parameter. If it is a positive value, then it means zoom in, and the fifth parameter indicates the PTZ channel number it linked with.</p> <p>"None" can be empty or none, which is used for compatibility.</p>	
+++PtzLinkEnable	bool	O	Enable PTZ linkage	true
+++SnapshotPeriod	int	O	Frame interval. It means the number of frames it takes to capture one image, and in a certain period of time, the number of images depends on the capturing frame rate. 0 indicates it takes multiple snapshots continuously.	3
+++SnapshotTimes	int	O	Times of continuous snapshot. When SnapshotEnable is true, snapshotTimes=0 means the device will continuously take snapshots until the event ends.	5

+++VoiceEnable	bool	O	Voice prompts	true
+++Voice	object	O	Linkage prompted by voice	
++++Channels	int[]	O	The channel number that the audio prompts playing.	[1, 2]
++++AudioFileName	char[256]	O	Absolute path of the linkage audio file. It can be an empty string. If it is, it means the default alarm linkage voice that specified by Sound. AlarmSound. "Null" means no audio linkage.	"/mnt/alarm.g711a"
++++PlayTimes	uint	O	Playing times of the linkage audio.	3
++++LinkTime	uint	O	The playing time of the linkage audio. Unit: second	30
++++Interval	uint	O	The playing interval of the sound, and the default value is 0. Unit: second.	30
++Config	object	O	The specific configuration of rule, and the corresponding configuration depends on the value of Type. For more information, please see AudioAnalyseRule(<RuleType>).	
+Global	object	O	Global configuration. This field has no relationship with PTZ position.	
++ClassList	char[32][64]	O	The valid intelligent business	["<ClassType1>","<ClassType2>"]
++Module	object	O	Business configuration.	

			The field name indicates the business name. For the detailed configuration, please see AudioAnalyseModule(<ClassType>).	
+++SmartSoundDetect	object	O	Sound detection business	
++++Thresh old	uint32	O	Sound intensity threshold, and it ranges from [1,100]	50
++Detail	object	O	Other universal or non-service related global configurations.	
+Scene	object[]	O	Scene configuration	
++PresetID	uint	O	Preset ID	0
++ClassList	char[32][64]	O	Valid intelligent business	"<ClassType1>","<ClassType2>"
++Module	object	O	Refer to the module under“Global”	
++Detail	object	O	Other universal or non-service related scene configurations.	
+State	uint32	O	Analysis status of audio channel 0 : not support 1 : support	1
Complete Example				
{ "Config": { "AudioChannel": 0, "Global": { "ClassList": [], "Detail": null, "Module": null }, "Rules": [{ "Class": "SmartSoundDetect", "Config": { 				

```

        "Sensitive": 5,
        "TimesThreshold": 2,
        "Types": [
            "AudioCrashingGlass",
            "AudioScream",
            "AudioGunshot",
            "AudioExplosion"
        ],
    },
    "Enable": false,
    "EventHandler": {
        "TourEnable": false,
        "VoiceEnable": false,
        ...
    },
    "Id": 1,
    "Name": "SmartSoundDetect",
    "PresetID": 0,
    "Type": "SmartSoundDetect"
}
],
"Scene": [
{
    "ClassList": [],
    "Detail": null,
    "Module": {
        "SmartSoundDetect": {
            "Threshold": 50
        }
    },
    "PresetID": 0
}
],
"State": 1
}
}

```

4.3.7 Setting the Analysis Configuration of the Specified Audio Channel

Configure the analysis configuration of the specified audio channel. We recommend you to use it in combination with getConfig. <http://<server>/cgi-bin/api/AudioAnalyseManager/getConfig>

Request URL	<a href="http://<server>/cgi-bin/api/AudioAnalyseManager/setConfig">http://<server>/cgi-bin/api/AudioAnalyseManager/setConfig
--------------------	---

Method	POST			
Parameter Format	key=value format at URL			
Name	Type	Required	Description	Example
Config	object	R	<p>For the detailed content, please see getConfig to request for the returned value.</p> <p>http://<server>/cgi-bin/api/AudioAnalyseManager/getConfig</p>	
AudioChannel	int32	O	<p>Audio channel number. When there are no signs indicate that the audio source is from the Internet, then the AudioChannel means the local channel number. When using the number, it must be linked with logical audio channel through channel number or other fields. If fails to link, the API will be failed to return.</p>	0
DeviceID	char[64]	O	<p>Device ID. It mainly used in the back-end cascade scenes, and if there are no special requirements, it will not be used in front-end devices.</p>	"12345678"
Complete Example				
<pre>{ "AudioChannel": 0, "DeviceID": "12345678", "Config": { "AudioChannel": 0, "Global": { "ClassList": [], "Detail": null, "Module": null }, "Rules": [{ </pre>				

```

    "Class": "SmartSoundDetect",
    "Config": {
        "Sensitive": 6,
        "TimesThreshold": 1,
        "Types": [
            "AudioCrashingGlass",
            "AudioScream",
            "AudioGunshot",
            "AudioExplosion"
        ]
    },
    "Enable": false,
    "EventHandler": {
        "TourEnable": false,
        "VoiceEnable": false,
        ...
    },
    "Id": 1,
    "Name": "SmartSoundDetect",
    "PresetID": 0,
    "Type": "SmartSoundDetect"
}
],
"Scene": [
{
    "ClassList": [],
    "Detail": null,
    "Module": {
        "SmartSoundDetect": {
            "Threshold": 50
        }
    },
    "PresetID": 0
}
],
"State": 1
}
}

```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

4.3.8 Getting the Related Business Configuration of the Specified Audio Channel

Get the relating business configuration of the specified audio channel.

Request URL	http://<server>/cgi-bin/api/AudioAnalyseManager/getClassConfig			
Method	POST			
Parameter Format	key=value format at URL			
Name	Type	Required	Description	Example
Class	char[64]	R	Audio analysis business	"<ClassType>"
PresetID	int32	O	Preset. If it is empty, the default value 0 is taken and the field can be empty for IPC.	0
AudioChannel	int32	O	Audio channel number. When there are no signs indicating that the audio source is from the Internet, then the AudioChannel means the local channel number. When using the number, it must be linked with logical audio channel through channel number or other fields. If it fails to link, the API will be failed to return.	0
DeviceID	char[128]	O	Device ID. It mainly used in the back-end cascade scenes, and if there are no special requirements, it will not be used in front-end devices.	"12345678"
Complete Example				
<pre>{ "Class": "SmartSoundDetect", "PresetID": 0, "AudioChannel": 0, "DeviceID": "12345678" }</pre>				

```
}
```

Response Params (JSON format in body)				
Name	Type	Required	Description	Example
Config	object	O		
+Rules	object[]	O		
++Class	char[64]	O	Rule category	"<ClassType>"
++Type	char[64]	O	Rule type	"<RuleType>"
++Id	uint	O	Rule ID. It can be 0, which means the system will automatically allocate the appropriate ID. If there are businesses that bonds the rule ID, the client will be responsible for ensuring that the code and the name are unique.	0
++Name	char[64]	O	Rule name	"..."
++Enable	bool	O	Enable the rules	true
++PresetID	uint32	O	Preset spots	0
++EventHandler	object	O	Rule linkage Items	
+++TimeSection	char[8][6][20]	O	Alarm period	[["1 00:00:00-08:00:00", "1 10:00:00-24:00:00",]]
+++RecordEnable	bool	O	Enable recording, with RecordChannels. Record is enabled if it is true; start recording when the event action is start and stop recording when it is stop. Record is disabled if it is false.	true
+++RecordChannels	int[]	O	Recording channel number list One-dimensional array. Each	[0, 1, 2]

			member indicates that the corresponding channel needs to record, and the channel number starts from 0.	
+++RecordLatch	int	O	Recording delay time (second) Range [10, 300]	10
+++AlarmOutputEnable	bool	O	Enable alarm output	true
+++AlarmOutputChannels	int[]	O	Alarm output channel number list One-dimensional array. Each member indicates that the corresponding channel needs to output alarm, and the channel number starts from 0.	[1, 4]
+++AlarmOutputLatch	int	O	Output delay time (second) after the alarm output stops. Scope [1,300]	10
+++SnapshotEnable	bool	O	Enable snapshot	true
+++SnapshotChannels	int[]	O	Snapshot channel number list One-dimensional array. Each member indicates that the corresponding channel needs to take snapshot, and the channel number starts from 0.	[2, 4]
+++MailEnable	bool	O	Send emails. If there are images, the images will be sent as email attachment.	true
+++Dejitter	int	O	Dejitter time. The field is only meaningful for part of the start or stop type events. Unit: Second (0 s–600 s). It can only be used for applications to automatically tour the detected events, such as motion detection and tampering. The similar intelligent events that	0

			reported by the algorithm database such as tripwire and face detection cannot configure dejitter time.	
+++BeepEnable	bool	O	Buzzer:	true
+++Delay	uint	O	Set the delay time for the event to take effect. Unit: second [0-300].	
+++ExAlarmOutChannels	int[]	O	Expands alarm output channel list One-dimensional array. Each member indicates that the corresponding channel needs to output extension alarm, and the channel number starts from 0.	[2, 3]
+++ExAlarmOutEnable	bool	O	Enable alarm output extension	true
+++LightingLink	object	O	PTZ illuminator linkage item	
++++Enable	bool	O	Enable	true
++++FilckerLightType	enumchar[32]	O	Flicker light type enumchar[32]{ "WhiteLight": indicates the flashing light is white light "RedBlueLight": indicates the flashing light is red and blue light }	"WhiteLight"
++++LightLinkType	enumchar[32]	O	Light linkage way enumchar[32]{ "Filcker": flicker (by default) "KeepLighting": solid on }	"Filcker"
++++FilckerIntervalTime	float	O	Flicker interval Unit: 0.1s, and the default value is 0.5 s.	5
++++FilckerTimes	int	O	The flicker times that can be configured. Unit: times, and the default	5

			value is 5.	
++++LightDuration	uint	O	The duration time of the flicker or normally lighting. Unit: second.	10
++++LightBright	uint	O	Brightness of the linked light The range refers to the capacity of LinkLightBrightRange. If this item does not exist, the default brightness will be adopted.	50
+++LogEnable	bool	O	Whether to record alarm logs. The type is fixed to EventStart/EventStop/EventPulse and the original event type is recorded in the Detail.Code field in the log.	false
+++LogRemote	bool	O	Whether to record remote logs. The type is fixed as EventStart/EventStop/EventPulse and the original event type is recorded in the Detail.Code field in the log. (When LogRemote and LogEnable exist at the same time, the remote log records will use the data from LogRemote).	false
+++MMSEnable	bool	O	Enable sending MMS	false
+++MatrixChannels	int[]	O	Link with video matrix channels number	[1, 6]
+++MatrixEnable	bool	O	Enable linkage video matrix	true
+++MessageEnable	bool	O	Whether to upload the message to the alarm central server. Alarm center does not need to log in to the device, and the device itself can directly send the message to alarm center.	true

+++PtzLink	object[][]	O	<p>PTZ linkage. It is a two-dimensional array, each member indicates a linkage item, and the last item indicates channel number. Each linkage item includes 5 components including linkage type, three relating parameters and the PTZ channel it linked with.</p> <p>Linkage types include:</p> <ul style="list-style-type: none"> "Preset": linkage preset; "Tour": linkage tour ; "Pattern" : linkage mode. "Zoom" : linkage zoom "SingleScene": link with single scene. (The second linkage parameter indicates the scene number, and the third linkage parameter indicates the stay time of the scene after the alarm finished, the fourth linkage parameter indicates reserve the parameter and the fifth linkage parameter indicates the PTZ channel it linked with. "QuickFocus": the thermal imaging PTZ links with quick focus function (the second one indicates the zoom of the thermal imaging channel), and its value ranges from 0 to 24. The third parameter indicates the stay time and the fourth indicates reserve the parameter. If it is a positive value, then it means zoom in, and the fifth parameter indicates the PTZ channel number it linked with. "None" can be empty or none, which is used for 	[["Preset", 3, 0, 0, 0], ["Tour", 4, 0, 0, 1], ["Pattern", 1, 0, 0, 2]]
------------	------------	---	---	---

			compatibility.	
+++PtzLinkEnable	bool	O	Enable PTZ linkage	true
+++SnapshotPeriod	int	O	Frame interval. It means the number of frames it takes to capture one image, and in a certain period of time, the number of images depends on the capturing frame rate. 0 indicates it takes multiple snapshots continuously.	3
+++SnapshotTimes	int	O	Times of continuous snapshot. When SnapshotEnable is true, snapshotTimes=0 means the device will continuously take snapshots until the event ends.	5
+++VoiceEnable	bool	O	Voice prompts	true
+++Voice	object	O	Audio Linkage	
++++Channels	int[]	O	Channel	[1, 2]
++++AudioFileName	char[256]	O	Absolute path of the linkage audio file. It can be an empty string. If it is, it means the default alarm linkage voice that specified by Sound. AlarmSound. "Null" means no audio linkage.	"/mnt/alarm.g711a"
++++PlayTimes	uint	O	Play times of the linkage audio.	3
++++LinkTime	uint	O	The playing time of the linkage audio. Unit: second	30
++++Interval	uint	O	The playing interval of the sound, and the default value is 0. Unit: second.	30
++Config	object	O	The specific configuration of rule, and the corresponding configuration depends on the	

			value of Type. For more information, please see AudioAnalyseRule(<RuleType>).	
+Module	object	O	The specific business configuration, and its value refer to the "request" section. For more information, please see AudioAnalyseModule(<Class Type>)	
+Global	object	O		
++Detail	object	O	Universal configuration of the global business	
+Scene	object	O	The business universal configuration of the current preset	
++PresetID	uint32	O	Preset ID	0
++Detail	object	O		
Complete Example				
<pre>{ "Config": { "Global": { "Detail": null }, "Module": { "Threshold": 50 }, "Rules": [{ "Class": "SmartSoundDetect", "Config": { "Sensitive": 6, "TimesThreshold": 1, "Types": ["AudioCrashingGlass", "AudioScream", "AudioGunshot", "AudioExplosion"] }, "Enable": false, }] } }</pre>				

```

    "EventHandler": {
        "TourEnable": false,
        "VoiceEnable": false,
        ...
    },
    "Id": 1,
    "Name": "SmartSoundDetect",
    "PresetID": 0,
    "Type": "SmartSoundDetect"
}
],
"Scene": {
    "Detail": null
}
}
}

```

4.3.9 Configuring the Related Business Configuration of the Specified Audio Channel

Configure the analysis configuration of the specified audio channel. We recommend you to use it in combination with getConfig. <http://<server>/cgi-bin/api/AudioAnalyseManager/getConfig>

Request URL	http://<server>/cgi-bin/api/AudioAnalyseManager/setClassConfig			
Method	POST			
Parameter Format	key=value format at URL			
Name	Type	Required	Description	Example
Configuration	object	R	For the detailed content, please see getConfig to request for the returned value. <a href="http://<server>/cgi-bin/api/AudioAnalyseManager/getConfig">http://<server>/cgi-bin/api/AudioAnalyseManager/getConfig	
Class	char[64]	R	Audio analysis business	"<ClassType>"
AudioChannel	int32	O	Audio channel number. When there are no signs indicate that the audio source is from the Internet, then the AudioChannel means the local channel	0

			number. When using the number, it must be linked with logical audio channel through channel number or other fields. If fails to link, the API will be failed to return.	
DeviceID	char[128]	O	Device ID. It is mainly used in the back-end cascade scenes, and if there are no special requirements, it will not be used in front-end devices.	"12345678"
PresetID	int32	O	Preset. When there are no fields, the default preset is 0, and the field can be empty for IPC.	0
Complete Example				
<pre>{ "Config": { "Global": { "Detail": null }, "Module": { "Threshold": 50 }, "Rules": [{ "Class": "SmartSoundDetect", "Config": { "Sensitive": 6, "TimesThreshold": 1, "Types": ["AudioCrashingGlass", "AudioScream", "AudioGunshot", "AudioExplosion"] }, "Enable": false, "EventHandler": { "TourEnable": false, "VoiceEnable": false, ... }, ... }] } }</pre>				

```

        "Id": 1,
        "Name": "SmartSoundDetect",
        "PresetID": 0,
        "Type": "SmartSoundDetect"
    }
],
"Scene": {
    "Detail": null
}
},
"Class": "SmartSoundDetect",
"AudioChannel": 0,
"DeviceID": "12345678",
"PresetID": 0
}

```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

4.3.10 AudioAnalyseRule Config

4.3.10.1 AudioAnalyseRule(SmartSoundDetect)

Name	Type	R/O	Description	Example
Types	char[32][32]	R	Sound Type enumchar[][][32]{ "AudioCry" "AudioAlarm" "AudioGunshot" "AudioExplosion" "AudioScream" "AudioCrashingGlass" }	["AudioCry"]
Sensitive	uint32	O	Detection sensitivity, range [1,10]	5
TimesThreshold	uint32	O	Times threshold, range [1, 5]	2

4.3.11 AudioAnalyseModule Config

4.3.11.1 AudioAnalyseModule(SmartSoundDetect)

Name	Type	R/O	Description	Example

Threshold	uint32	O	Sound intensity threshold, range [1, 100]	50
-----------	--------	---	---	----

4.3.12 Play the Specified Audio in the Device

Request URL	http://<server>/cgi-bin/api/Speak/startPlayByTime			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Path	string	R	Audio file path	"/Audio_Path"
FileName	string	R	Audio file name	"alarm.wav"
PlayTimes	int32	O	times of broadcasts, default once	1
Request Example				
<pre>{ "Path": "/Audio_Path", "FileName": "alarm.wav", "PlayTimes": 1 }</pre>				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Status	enumchar r[32]	R	enumchar[32]{ "OK": indicates that the operation is successful and returns before the end of the broadcast, which is different from API "PostAudio" "Busy": Indicates that the device is broadcasting "Error": Broadcast error }	"OK"
Response Example				
<pre>{ "Status" : "OK" }</pre>				

4.4 Snapshot

4.4.1 [Config] Snap

Snapshot configuration parameters:

Config Data Params

Name	Type	R/O	Description	Example
Snap	object[]	R	One element for each channel, and the array index represents the channel number which starts from 0.	[80,50]
+HolidayEnable	bool	O	Whether holiday setting is supported. It is not supported by default.	false
TimeSection	char[8][24][32]		<p>Configure the period of capturing snapshots.</p> <p>Two-dimensional array in strings.</p> <p>The first dimension of the array is the day of the week, and the range is [0–7] (Sunday–Saturday). The last item is for holiday.</p> <p>The second dimension is the period index. One day is divided into multiple periods, and the range is [0–23].</p> <p>Each period is set as a string in the format of:mask hh: mm: ss-hh: mm: ss</p> <p>Mask: The range is [0–4294967295], and each bit carries a special meaning. See the description as follows:</p> <ul style="list-style-type: none"> hh: Hour; range: [0–24] mm: Minute; range: [0–59] ss: Description; ranges [0 to 59] <p>Mask: record type.</p> <p>Meaning of each bit:</p> <ul style="list-style-type: none"> Bit0: Normal snapshot Bit1: Motion detection snapshot Bit2: Alarm snapshot Bit3: Card-swiping snapshot Bit6: POS snapshot Bit4, Bit5, Bit7–Bit31: Reserved. 	[["6 00:00:00-23:59:59", "0 00:00:00-23:59:59", "0 00:00:00-23:59:59", ...]]

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

<http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=Snapshot>

Get Config Response Example

table.Snap[0].HolidayEnable=true

table.Snap[0].TimeSection[0][0]=6 00:00:00-23:59:59

table.Snap[0].TimeSection[0][1]=0 00:00:00-23:59:59

1

table.Snap[1].HolidayEnable=false

table.Snap[1].TimeSection[0][1]=0 00:00:00-23:59:59

table.Snap[1].TimeSection[0][1]=0 00:00:00-23:59:59

1

Set Config Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&Snap[0].HolidayEnable=true&Snap[0].TimeSection[0][0]=6 00:00:00-23:59:59&Snap[0].TimeSection[0][1]=0 00:00:00-23:59:59&Snap[1].HolidayEnable=false&Snap[1].TimeSection[0][1]=0 00:00:00-23:59:59&Snap[1].TimeSection[0][1]=0 0:00:00-23:59:59
```

Set Config Response Example

OK

4.4.2 Get a Snapshot

Get snapshots of a specified video channel

Request URL	http://<server>/cgi-bin/snapshot.cgi		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	Video channel number starting from 1, and the default value is 1.
type	uint32	O	0: Acquire the snapshot from the front end 1: Represent capture of local real-time video secondary decoding encoding Default 0
Request Example			
http://192.168.1.250/cgi-bin/snapshot.cgi?channel=1&type=0			

Response Params (binary in body)

<binary data>: images in JPEG format

Response Example

```
HTTP/1.1 200 OK
Server: Device/1.0
Content-Type: image/jpeg
Content-Length:<image size>

<jpeg data>
```

4.4.3 Subscribe to Snapshot

Subscribe to the events and event snapshots. For details on the event name and parameters, see the corresponding [Event] sections.

Request URL	http://<server>/cgi-bin/snapManager.cgi?action=attachFileProc		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	Video channel number, starting from 1. If not specified, the default value is 1. -1 means subscribing to all channels.
heartbeat	int	O	Heartbeat interval.

			<p>Unit: s Range: [1–60], and the default value is 5. While sending the events, the device regularly sends a heartbeat message to keep itself alive. The message is a string named "heartbeat".</p>	
Flags	char[][32]	R	<p>Subscription parameter. The value "Event" should be included.</p>	"Event"
Events	char[][32]	R	<p>The list of event codes Select All to subscribe all events Common event codes are as follows: VideoMotion: Motion detection VideoLoss: Video loss VideoBlind: Video Tampering AlarmLocal: Local alarms TrafficJunction: ANPR TrafficRetrograde: Wrong-way driving TrafficParking: Illegal parking TrafficJam: Traffic congestion TrafficThrow: Throwing out of the vehicle TrafficPedestrian: A pedestrian walks onto a motor vehicle lane TrafficParkingSpaceParking: A vehicle is in the parking space TrafficParkingSpaceNoParking: No vehicle is in the parking space TrafficParkingSpaceOverLine: A vehicle crosses the line of a parking space TrafficManualSnap: Manual snapshot capturing FaceRecognition: Face recognition TrafficParkingStatistics: Parking space counting AccessControl: Access control unlocking ...</p>	["VideoMotion" , "TrafficJunction"]

Request Example

```
http://192.168.1.250/cgi-bin/snapManager.cgi?action=attachFileProc&channel=1&heartbeat=5&Flags[0]=Event&Events=[VideoMotion%2CVideoLoss]
```

Response Params (multipart in body)				
Name	Type	R/O	Description	Example
Events	object[]	R	Event detail array	
Channel	int	O	The video channel number of the generated picture, starting from 0.	0
+EventBaseInfo	object	R	Shared basic information for all events	
++Code	char[32]	R	Event code	""FaceReco

			See the description on Events	gnition™
++Action	char[16]	R	Event action Value: "Start", "Stop" and "Pulse"	"Pulse"
++Index	int		The channel number where the event happens, starting from 0	0
+...	Unique parameters of each event For details on each event, see the corresponding section.	

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: closed

--<boundary>

Content-Type: text/plain

Content-Length: <data length>

Events[0].Channel=0

Events[0].EventBaseInfo.Code=TrafficJunction

Events[0].EventBaseInfo.Action=Pulse

Events[0].EventBaseInfo.Index=0

Events[0].CountInGroup=3

Events[0].IndexInGroup=1

Events[0].Lane=1

Events[0].PTS=42949485818.0

Events[0].TrafficCar.PlateNumber=ZZZ12345

Events[0].TrafficCar.DeviceAddress=XXRoad

.....

Events[1].Channel=1

Events[1].EventBaseInfo.Code=TrafficJunction

Events[1].EventBaseInfo.Action=Pulse

Events[1].EventBaseInfo.Index=1

Events[1].CountInGroup=3

Events[1].IndexInGroup=2

Events[1].Lane=1

.....

--<boundary>

Content-Type: image/jpeg

Content-Length:<image size>

<JPEG image data>

--<boundary>

Content-Type: text/plain

Content-Length:<data length>

Heartbeat

```
--<boundary>
```

```
.....
```

4.5 Video Attributes

4.5.1 Get Max Extra Stream Numbers

Get the maximum number of sub streams

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=getProductDefinition		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
name	char[]	R	Name Fixed value: MaxExtraStream
Request Example			
http://192.168.1.250/cgi-bin/magicBox.cgi?action=getProductDefinition&name=MaxExtraStream			

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
table	object	R	Return content	
+MaxExtraStream	int	R	The maximum number of sub streams Values: 1, 2 and 3.	1
Response Example				
table.MaxExtraStream=1				

4.5.2 Get Encode Capability

Get encoding capabilities.

Request URL	http://<server>/cgi-bin/encode.cgi?action=getCaps		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.250/cgi-bin/encode.cgi?action=getCaps			

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
caps	object	R	Encoding capability	
+MaxExtraStream	int	O	Maximum number of sub stream channels	3
+PreviewSplitNumList	int[]	O	The live view page can be split to multiple windows	[1, 4, 9, 16]
+PlaybackCompressedSplitNumList	int[]	O	Secondary compressed playback supports being split to multiple windows.	[1, 4, 9, 16]
+PreviewMode	char[16]	O	Multi-window live view mode. Values:	"SplitEncode"

			SplitEncode: Split encoding mode SnapShot: Fast snapshot mode SplitSnap: Split fast snapshot mode If the field does not exist or is null, it means the device does not support multi-window live view.	
+VideoEncodeDevices	object[]	O	Encoding capability Each element of the array represents a channel.	
++SupportIndividualResolution	bool	O	Whether independent resolution for snapshots is supported	true
++RecordIndividualResolution	bool	O	Whether independent resolutions for motion detection and alarm videos are supported.	true
++MaxCIFPFrameSize	int	O	CIF maximum value of the P frame. Unit: Kbits	40
++MinCIFPFrameSize	int	O	CIF minimum value of the P frame. Unit: Kbits	7
+CoriaDisplay	object[]	O	Display capability of mixed exposure. Each element of the array represents a channel.	
++CoriaDisplaySwitchs	int[]	O	Supported display mode: 0: Default 1: Vehicle mode 2: Person mode	[0,1,2]

Response Example

```

caps.MaxExtraStream=3
caps.PreviewSplitNumList[0]=1
caps.PreviewSplitNumList[1]=4
caps.PreviewSplitNumList[2]=9
caps.PlaybackCompressSplitNumList[0]=1
caps.PlaybackCompressSplitNumList[1]=4
caps.PlaybackCompressSplitNumList[2]=9
caps.PreviewMode=SplitSnap
caps.VideoEncodeDevices[0].SupportIndividualResolution=true
caps.VideoEncodeDevices[0].RecordIndividualResolution=true
caps.VideoEncodeDevices[0].MaxCIFPFrameSize=40
caps.VideoEncodeDevices[0].MinCIFPFrameSize=7
caps.CoriaDisplay[0].CoriaDisplaySwitchs[0]=0
caps.CoriaDisplay[0].CoriaDisplaySwitchs[1]=1
caps.CoriaDisplay[0].CoriaDisplaySwitchs[2]=2

```

4.5.3 Get Encode Config Capability

Get encoding configuration capabilities

Request URL	http://<server>/cgi-bin/encode.cgi?action=getConfigCaps
Method	GET
Request Params (key=value format in URL)	

Name	Type	R/O	Description	Example
channel	int	O	Video channel number. It is an integer starting from 1. When the channel does not exist, the default number is 1.	
Encode	object[]	O	The encoding configurations set before will influence the encoding configurations of other channels and streams.	
+MainFormat	object[]	O	Main stream. Each element of the array represents a recording type. 0: General encoding 1: Motion detection encoding 2: Alarm encoding	
++Video	object	O	Video format	
+++Compression	char[16]	O	Video encoding mode For example, "MPEG4", "H.264", "H.265"	"H.264"
+++Width	int	O	Video width Unit: pixels	1920
+++Height	int	O	Video height Unit: pixels	1080
+++BitRate	int	O	Video stream Unit: kbps	2048
+ExtraFormat	object[]	O	Sub stream Each element of the array represents a sub stream. 0: Sub stream 1 1: Sub stream 2 2: Sub stream 3	
++Video	object	O	Video format	
+++Compression	char[16]	O	Video encoding mode For example, "MPEG4", "H.264", "H.265"	"H.264"
+++Width	int	O	Video width Unit: pixels	720
+++Height	int	O	Video height Unit: pixels	576
+++BitRate	int	O	Video stream Unit: kbps	512

Request Example

`http://192.168.1.250/cgi-bin/encode.cgi?action=getConfigCaps&channel=1&Encode[0].MainFormat[0].Video.Compression=H.264&Encode[0].MainFormat[0].Video.Width=1920&Encode[0].MainFormat[0].Video.Height=1080`

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
caps	object	R	Encoding configuration capability	

+MainFormat	object[]	O	Main stream. Each element of the array represents a recording type. 0: General encoding 1: Motion detection encoding 2: Alarm encoding	
++Audio	object	O	Audio encoding capability	
+++CompressionTypes	char[]	O	Supported audio encoding formats: PCM, ADPCM, G.711A, G.711Mu, G.726, G.729, MPEG2, AMR, and SVAC.	G.711A, AMR
++Video	object	O	Video encoding capability	
+++CompressionTypes	char[]	O	Supported video encoding formats: MPEG4, MPEG2, MPEG1, MJPG, H.263, H.264 and H.265.	MPEG4, H.264, H.265
+++ResolutionTypes	char[]	O	Supported video resolutions. See the table below for the optional values.	D1, 1920x1080
+++BitRateOptions	char[]	O	Stream capability Two numbers are divided by a comma. The former number represents the minimal stream, and the latter represents the maximum stream. Unit: kbps	80,448
+++FPSMax	int	O	Maximum frame rate	25
+ExtraFormat	object[]	O	Sub stream. Each element of the array represents a sub stream. 0: Sub stream 1 1: Sub stream 2 2: Sub stream 3	
Audio In	object	O	Audio encoding capabilities	
+++CompressionTypes	char[]	O	Supported audio encoding formats: PCM, ADPCM, G.711A, G.711Mu, G.726, G.729, MPEG2, AMR, and SVAC.	G.711A, AMR
++Video	object	O	Video encoding capability	
+++CompressionTypes	char[]	O	Supported video encoding formats: MPEG4, MPEG2, MPEG1, MJPG, H.263, H.264 and H.265.	MPEG4, H.264, H.265
+++ResolutionTypes	char[]	O	Supported video resolutions. See the table below for the options.	D1, 1280x720
+++BitRateOptions	char[]	O	Stream capability Two numbers are divided by a comma. The former number represents the minimal stream, and the latter represents the maximum stream. Unit: kbps	80,448
+++FPSMax	int	O	Maximum frame rate:	25
+SnapFormat	object[]	O	Snapshot encoding capability Each element of the array represents a	

			snapshot capturing type. 0: General snapshot 1: Motion detection snapshot 2: Alarm snapshot	
++Video	object	O	Snapshot format	
+++CompressionTypes	char[16]	O	Supported encoding formats For example, "JPEG"	JPEG
+++ResolutionTypes	int	O	Supported video resolutions See the table below for the options.	D1, 1280x720
+++FPSMax	int	O	Maximum frame rate	2

Response Example

```

caps.MainFormat[0].Audio.CompressionTypes=G.711A, AMR
caps.MainFormat[0].Video.CompressionTypes=H.264,MJPG
caps.MainFormat[0].Video.ResolutionTypes=2048 x 1536,1080,SXGA, 1280 x 960,720,D1,CIF
caps.MainFormat[0].Video.BitRateOptions=448,2560
caps.MainFormat[0].Video.FPSMax=25
caps.ExtraFormat[0].Audio.CompressionTypes=G.711A, AMR
caps.ExtraFormat[0].Video.CompressionTypes=H.264,MJPG
caps.ExtraFormat[0].Video.ResolutionTypes=2048 x 1536,1080,SXGA, 1280 x 960,720,D1,CIF
caps.ExtraFormat[0].Video.BitRateOptions=448,2560
caps.ExtraFormat[0].Video.FPSMax=25
caps.SnapFormat[0].Video.CompressionTypes=JPEG
caps.SnapFormat[0].Video.ResolutionTypes=2048 x 1536,1080,SXGA, 1280 x 960,720,D1,CIF
caps.SnapFormat[0].Video.FPSMax=2

```

Appendix A: Video Resolution

Fixed Resolution Name	Size in PAL	Size in NTSC
"D1"	704 x 576	704 x 480
"HD1"	352 x 576	352 x 480
"BCIF"/"2CIF"	704 x 288	704 x 240
"CIF"	352 x 288	352 x 240
"QCIF"	176 x 144	176 x 120
"NHD"	640 x 360	—
"VGA"	640 x 480	—
"QVGA"	320 x 240	—
"SVCD"	480 x 480	—
"QQVGA"	160 x 128	—
"SVGA"	800 x 592	—
"SVGA1"	800 x 600	—
"WVGA"	800 x 480	—
"FWVGA"	854 x 480	—
"DVGA"	960 x 640	—
"XVGA"	1024 x 768	—
"WXGA"	1280 x 800	—
"WXGA2"	1280 x 768	—
"WXGA3"	1280 x 854	—

Fixed Resolution Name	Size in PAL	Size in NTSC
"WXGA4"	1366 x 768	—
"SXGA"	1280 x 1024	—
"SXGA+"	1400 x 1050	—
"WSXGA"	1600 x 1024	—
"UXGA"	1600 x 1200	—
"WUXGA"	1920 x 1200	—
"ND1"	240 x 192	—
"720P"	1280 x 720	—
"1080P"	1920 x 1080	—
"QFHD"	3840 x 2160	—
"1_3M", "1280x960"	1280 x 960 (1.3 Mega Pixels)	—
"2_5M", "1872x1408"	1872 x 1408 (2.5 Mega Pixels)	—
"5M", "3744x1408"	3744 x 1408 (5 Mega Pixels)	—
"3M", "2048x1536"	2048 x 1536 (3 Mega Pixels)	—
"5_0M", "2432x2048"	2432 x 2048 (5 Mega Pixels)	—
"1_2M", "1216x1024"	1216 x 1024 (1.2 Mega Pixels)	—
"1408x1024"	1408 x 1024 (1.5 Mega Pixels)	—
"3296x2472"	3296 x 2472 (8 Mega Pixels)	—
"5_1M", "2560x1920"	2560 x 1920 (5 Mega Pixels)	—
"960H",	960 x 576	960 x 480
"DV720P"	960 x 720	—
"2560x1600"	2560 x 1600 (4 Mega Pixels)	—
"2336x1752"	2336 x 1752 (4 Mega Pixels)	—
"2592x2048"	2592 x 2048	—
"2448x2048"	2448 x 2048	—
"1920x1440"	1920x1440	—
"2752x2208"	2752x2208	—
"3840x2160"	3840x2160	—
"4096x2160"	4096x2160	—
"3072x2048"	3072x2048	—

4.5.4 [Config] Encode of Media

- Get encode config

Table 4-1

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=Encode				
Method	GET				
Description	Get video encode config.				
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=Encode				
Success Return	<table border="0"> <tr><td>table.Encode[0].MainFormat[0].Audio.Bitrate=64</td></tr> <tr><td>table.Encode[0].MainFormat[0].Audio.Channels[0]=0</td></tr> <tr><td>table.Encode[0].MainFormat[0].Audio.Compression=G.711A</td></tr> <tr><td>table.Encode[0].MainFormat[0].Audio.Depth=16</td></tr> </table>	table.Encode[0].MainFormat[0].Audio.Bitrate=64	table.Encode[0].MainFormat[0].Audio.Channels[0]=0	table.Encode[0].MainFormat[0].Audio.Compression=G.711A	table.Encode[0].MainFormat[0].Audio.Depth=16
table.Encode[0].MainFormat[0].Audio.Bitrate=64					
table.Encode[0].MainFormat[0].Audio.Channels[0]=0					
table.Encode[0].MainFormat[0].Audio.Compression=G.711A					
table.Encode[0].MainFormat[0].Audio.Depth=16					

```

table.Encode[0].MainFormat[0].Audio.Frequency=8000
table.Encode[0].MainFormat[0].Audio.Mode=0
table.Encode[0].MainFormat[0].Audio.Pack=PS
table.Encode[0].MainFormat[0].AudioEnable=true
table.Encode[0].MainFormat[0].Video.resolution=1920x1080
table.Encode[0].MainFormat[0].Video.BitRate=4096
table.Encode[0].MainFormat[0].Video.BitRateControl=CBR
table.Encode[0].MainFormat[0].Video.Compression=H.264
table.Encode[0].MainFormat[0].Video.CustomResolutionName=1080P
table.Encode[0].MainFormat[0].Video.FPS=18
table.Encode[0].MainFormat[0].Video.GOP=36
table.Encode[0].MainFormat[0].Video.Height=1080
table.Encode[0].MainFormat[0].Video.Pack=PS
table.Encode[0].MainFormat[0].Video.Profile=High
table.Encode[0].MainFormat[0].Video.Quality=4
table.Encode[0].MainFormat[0].Video.QualityRange=6
table.Encode[0].MainFormat[0].Video.SVCTLayer=1
table.Encode[0].MainFormat[0].Video.Width=1920
table.Encode[0].MainFormat[0].Video.Priority=0
table.Encode[0].MainFormat[0].VideoEnable=true
table.Encode[0].MainFormat[1].Audio.Bitrate=64
table.Encode[0].MainFormat[1].Audio.Channels[0]=0
table.Encode[0].MainFormat[1].Audio.Compression=G.711A
table.Encode[0].MainFormat[1].Audio.Depth=16
table.Encode[0].MainFormat[1].Audio.Frequency=8000
table.Encode[0].MainFormat[1].Audio.Mode=0
table.Encode[0].MainFormat[1].Audio.Pack=PS
table.Encode[0].MainFormat[1].AudioEnable=true
table.Encode[0].MainFormat[1].Video.resolution=1920x1080
table.Encode[0].MainFormat[1].Video.BitRate=4096
table.Encode[0].MainFormat[1].Video.BitRateControl=CBR
table.Encode[0].MainFormat[1].Video.Compression=H.264
table.Encode[0].MainFormat[1].Video.CustomResolutionName=1080P
table.Encode[0].MainFormat[1].Video.FPS=18
table.Encode[0].MainFormat[1].Video.GOP=36
table.Encode[0].MainFormat[1].Video.Height=1080
table.Encode[0].MainFormat[1].Video.Pack=PS
table.Encode[0].MainFormat[1].Video.Profile=High
table.Encode[0].MainFormat[1].Video.Quality=4
table.Encode[0].MainFormat[1].Video.QualityRange=6
table.Encode[0].MainFormat[1].Video.SVCTLayer=1
table.Encode[0].MainFormat[1].Video.Width=1920
table.Encode[0].MainFormat[1].Video.Priority=0
table.Encode[0].MainFormat[1].VideoEnable=true
table.Encode[0].MainFormat[2].Audio.Bitrate=64
table.Encode[0].MainFormat[2].Audio.Channels[0]=0
table.Encode[0].MainFormat[2].Audio.Compression=G.711A

```

```

table.Encode[0].MainFormat[2].Audio.Depth=16
table.Encode[0].MainFormat[2].Audio.Frequency=8000
table.Encode[0].MainFormat[2].Audio.Mode=0
table.Encode[0].MainFormat[2].Audio.Pack=PS
table.Encode[0].MainFormat[2].AudioEnable=true
table.Encode[0].MainFormat[2].Video.resolution=1920x1080
table.Encode[0].MainFormat[2].Video.BitRate=4096
table.Encode[0].MainFormat[2].Video.BitRateControl=CBR
table.Encode[0].MainFormat[2].Video.Compression=H.264
table.Encode[0].MainFormat[2].Video.CustomResolutionName=1080P
table.Encode[0].MainFormat[2].Video.FPS=18
table.Encode[0].MainFormat[2].Video.GOP=36
table.Encode[0].MainFormat[2].Video.Height=1080
table.Encode[0].MainFormat[2].Video.Pack=PS
table.Encode[0].MainFormat[2].Video.Profile=High
table.Encode[0].MainFormat[2].Video.Quality=4
table.Encode[0].MainFormat[2].Video.QualityRange=6
table.Encode[0].MainFormat[2].Video.SVCTLayer=1
table.Encode[0].MainFormat[2].Video.Width=1920
table.Encode[0].MainFormat[2].Video.Priority=0
table.Encode[0].MainFormat[2].VideoEnable=true
table.Encode[0].MainFormat[3].Audio.Bitrate=64
table.Encode[0].MainFormat[3].Audio.Channels[0]=0
table.Encode[0].MainFormat[3].Audio.Compression=G.711A
table.Encode[0].MainFormat[3].Audio.Depth=16
table.Encode[0].MainFormat[3].Audio.Frequency=8000
table.Encode[0].MainFormat[3].Audio.Mode=0
table.Encode[0].MainFormat[3].Audio.Pack=PS
table.Encode[0].MainFormat[3].AudioEnable=true
table.Encode[0].MainFormat[3].Video.resolution=704x576
table.Encode[0].MainFormat[3].Video.BitRate=2048
table.Encode[0].MainFormat[3].Video.BitRateControl=VBR
table.Encode[0].MainFormat[3].Video.Compression=H.264
table.Encode[0].MainFormat[3].Video.FPS=25
table.Encode[0].MainFormat[3].Video.GOP=50
table.Encode[0].MainFormat[3].Video.Height=576
table.Encode[0].MainFormat[3].Video.Pack=PS
table.Encode[0].MainFormat[3].Video.Profile=Main
table.Encode[0].MainFormat[3].Video.Quality=4
table.Encode[0].MainFormat[3].Video.QualityRange=6
table.Encode[0].MainFormat[3].Video.SVCTLayer=1
table.Encode[0].MainFormat[3].Video.Width=704
table.Encode[0].MainFormat[3].Video.Priority=0
table.Encode[0].MainFormat[3].VideoEnable=true
table.Encode[0].ExtraFormat[0].Audio.Bitrate=64
...
table.Encode[0].SnapFormat[0].Audio.Bitrate=64

```

	...
Comment	<p>Parameters in Response:</p> <p>The format of the config is <i>head.configItems</i>. The <i>head</i> can be:</p> <p><i>headMain</i>= table. Encode[Channel].MainFormat[Type] <i>headSnap</i> = table. Encode[Channel].SnapFormat[Type] <i>headExtra</i> =table. Encode[Channel].ExtraFormat[ExtraStream]</p> <p><i>Channel</i>: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p> <p><i>Type</i>:</p> <ul style="list-style-type: none"> 0 = regular encode 1 = motion detection encode 2 = alarm encode 3= emergency encode <p><i>ExtraStream</i>:</p> <ul style="list-style-type: none"> 0 = extra stream 1 1 = extra stream 2 2 = extra stream 3 <p>The <i>configItems</i> are list as bellow.</p>

- Set encode config

Table 4-2

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&< paramName >=< paramValue >[&< paramName >=< paramValue >...]
Method	GET
Description	Set encode config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&Encode[1].MainFormat[0].Video.Compression=MPEG4&
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>In table below,</p> <p>head=Encode[Channel].MainFormat[RecordType] (or) Encode[Channel].ExtraFormat[ExtraStream]</p> <p><i>Channel</i>: integer, array index starts from 0, which means video channel>equals to video channel index -1, and so 0 means channel 1).</p> <p><i>RecordType</i>:</p> <ul style="list-style-type: none"> 0 = regular record 1 = motion detection record 2 = alarm record <p><i>ExtraStream</i>:</p> <ul style="list-style-type: none"> 0 = extra stream 1 1 = extra stream 2 2 = extra stream 3

Appendix A: Video Encode Config

ParamName	ParamValue type	Description
head.Video.BitRate	integer	Unit is Kbps Range depends on capability in GetVideoConfigCaps
head.Video.BitRateControl	string	Range is {CBR, VBR} CBR: constant bitrates VBR: variable bitrates
head.Video.Compression	String	Range is {MPEG4, MPEG2, MPEG1, MJPG, H.263, H.264, H.265, SVAC2} Depends on capacity in GetVideoConfigCaps
head.Video.FPS	float	Frames per second. <1.0: several seconds/frame, FPS=0.3333: 3 seconds per frame. >1.0: several frames/second. FPS=3: 3 frames per second. Range depends on capability in GetVideoConfigCaps
head.Video.GOP	integer	Range is [1—100]. Group of picture, it's the interval of I Frame, Example: GOP=50, means there is one I frame every 49 P or B frames
head.Video.Height	integer	Video height
head.Video.Width	integer	Video Width
head.Video.Profile	String	Range is { Baseline, Main , Extended , High } Only when video compression is H.264, it's effective.
head.Video.Quality	integer	Range is [1—6]. Image Quality, available when Video.BitRateControl=VBR 1: worst quality 6: best quality
head.Video.Priority	integer	Range is 0 or 1 0 for Image Quality first 1 for Bitstream first
head.VideoEnable	bool	True: enable video

Appendix B: Audio Encode Config

ParamName	ParamValue type	Description
head.Audio.Bitrate	integer	Unit is kbps Range depends on capacity in GetAudioConfigCaps
head.Audio.Compression	string	Range depends on capacity in GetAudioConfigCaps
head.Audio.Depth	integer	Audio sampling depth
head.Audio.Frequency	integer	Audio sampling frequency

ParamName	ParamValue type	Description
head.Audio.Mode	integer	Range is {0,1,2,3,4,5,6,7} Audio encode mode. 0: 4.75kbps, 1: 5.15 kbps, 2: 5.9 kbps, 3: 6.7 kbps, 4: 7.4 kbps, 5: 7.95 kbps, 6: 10.2 kbps, 7: 12.2 kbps,
head.AudioEnable	bool	Enable/Disable audio

4.5.5 [Config] Encode of Region Interested

- Get encode config of region interested

Table 4-3

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=VideoEncodeROI
Method	GET
Description	Get video encode config of region interested.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=VideoEncodeROI
Success Return	<pre> head.Regions[0][0]=0 head.Regions[0][1]=0 head.Regions[0][2]=0 head.Regions[0][3]=0 head.Regions[1][0]=0 ... head.Quality=4, head.Main=true, head.Extra1=true, head.Extra2=true, head.Extra3=true, head.Snapshot=true head.DynamicTrack=true, head.DynamicDelayTime=60 </pre>
Comment	<p>Parameters in Response:</p> <p>head=table.VideoEncodeROI[<i>ChannelNo</i>]</p> <p>ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p>

- Set encode config of region interested

Table 4-4

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramV>
--------	---

	alue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set video encode config of region interested.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoEncodeROI[0].DynamicTrack=true
Success Return	OK
Comment	<p>Parameters in URL: paramName and paramValue are as table below. In table below, head = VideoEncodeROI[ChannelNo] ChannelNo: integer, array index starts from 0, which means video channel>equals to video channel index -1, and so 0 means channel 1).</p>

Appendix:

ParamName	ParamValue type	Description
head . DynamicTrack	bool	Enable/Disable

4.5.6 [Config] Channel Title

Channel title configuration parameters:

Config Data Params				
Name	Type	R/O	Description	Example
ChannelTitle	object[]	R	Array One element for each channel, and the array index represents the channel number which starts from 0.	
+Name	char[]	R	The content represented in the form of string. '\r' means line change, and up to two lines are supported.	"CAM1 123"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example
http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=ChannelTitle
Get Config Response Example
table.ChannelTitle[0].Name=CAM1 123
table.ChannelTitle[1].Name=CAM2 456
...

Set Config Request Example
http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&ChannelTitle[0].Name=CAM1 123&ChannelTitle[1].Name=CAM2 456
Set Config Response Example
OK

4.5.7 Get Video Input Channels Device Supported

Table 4-5

Syntax	http://<server>/cgi-bin/devVideoInput.cgi?action=getCollect
Method	GET
Description	Get the video input channel numbers.
Example	http://192.168.1.250/cgi-bin/devVideoInput.cgi?action=getCollect
Success Return	result=1
Comment	—

4.5.8 Get Video Output Channels Device Supported

Table 4-6

Syntax	http://<server>/cgi-bin/devVideoOutput.cgi?action=getCollect
Method	GET
Description	Get the video output channel numbers.
Example	http://192.168.1.250/cgi-bin/devVideoOutput.cgi?action=getCollect
Success Return	result=2
Comment	—

4.5.9 Get Max Remote Input Channels

Table 4-7

Syntax	http://<server>/cgi-bin/magicBox.cgi?action=getProductDefinition&name=MaxRemoteInputChannels
Method	GET
Description	Get max remote input channels.
Example	http://192.168.1.250/cgi-bin/magicBox.cgi?action=getProductDefinition&name=MaxRemoteInputChannels
Success Return	table.MaxRemoteInputChannels=16
Comment	MaxRemoteInputChannels: max remote input channels.

4.5.10 [Config] Video Standard

- Get video standard

Table 4-8

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=VideoStandard
Method	GET

Description	Get video standard config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=VideoStandard
Success Return	table.VideoStandard=PAL
Comment	—

- Set video standard

Table 4-9

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&VideoStandard=< paramValue >
Method	GET
Description	Set video standard config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoStandard=PAL
Success Return	OK
Comment	VideoStandard: string, range is {PAL, NTSC} Video Standard.

4.5.11 [Config] Video Widget

- Get video widget config

Table 4-10

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=VideoWidget
Method	GET
Description	Video widget config contains Channel Title, User-defined Title, Covers and Time Title parameters, defines the background color, front color and positions of channel title and time title, User-defined title and defines the regions which are not visible (cover).
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=VideoWidget
Success Return	<pre> head.BackColor[0]=0 head.BackColor[1]=0 head.BackColor[2]=0 head.BackColor[3]=128 head.EncodeBlend=true head.FrontColor[0]=255 head.FrontColor[1]=255 head.FrontColor[2]=255 head.FrontColor[3]=0 head.Rect[0]=0 head.Rect[1]=8191 head.Rect[2]=0 head.Rect[3]=8191 ... </pre>
Comment	Parameters in Response: head =table.VideoWidget[Channel].ChannelTitle (or)

	<p>table.VideoWidget[Channel].Covers[CoReg] (or) table.VideoWidget[Channel].TimeTitle table.VideoWidget[Channel].UserDefinedTitle[Index]</p> <p>Channel: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). CoReg: Cover Region, Covers is an array which sustains multi- Cover regions 0 = region 1 1 = region 2 2 = region 3 3 = region 4 Index: integer, User-defined title index, now only index 0 is valid, others are reserved.</p>
--	--

- Set video widget config

Table 4-11

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&< paramName >=< paramValue >[&< paramName >=< paramValue >...]
Method	GET
Description	Set video widget config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoWidget[1].Covers[0].BackColor[0]=128&VideoWidget[1].Covers[0].BackColor[1]=128&VideoWidget[1].Covers[0].BackColor[2]=128&VideoWidget[1].Covers[0].BackColor[3]=0
Success Return	OK
Comment	<p>Parameters in URL: In table below,</p> <p>headChannelTitle = VideoWidget[Channel].ChannelTitle headCover = VideoWidget[Channel].Covers[CoReg] headTimeTitle = VideoWidget[Channel].TimeTitle headUserDefinedTitle = VideoWidget[Channel].UserDefinedTitle[Index]</p> <p>Channel: integer, array index starts from 0, which means video channel>equals to video channel index -1, and so 0 means channel 1). CoReg: Cover region index. Covers is an array which contains multiple cover regions 0 = region 1 1 = region 2 2 = region 3 3 = region 4 Index: integer, User-defined title index, now only index 0 is valid, others are reserved.</p>

Appendix:

ParamName	ParamValue type	Description
-----------	-----------------	-------------

ParamName	ParamValue type	Description
<code>headCover.BackColor[0]</code> <code>headCover.BackColor[1]</code> <code>headCover.BackColor[2]</code> <code>headCover.BackColor[3]</code>	integer	Range is [0—255]. BackColor[0]:red value BackColor[1]:green value BackColor[2]:blue value BackColor[3]: alpha value
<code>headCover.EncodeBlend</code>	bool	false - widget blend is not enabled.
<code>headCover.FrontColor[0]</code> <code>headCover.FrontColor[1]</code> <code>headCover.FrontColor[2]</code> <code>headCover.FrontColor[3]</code>	integer	Range is [0—255]. FrontColor[0]:red value FrontColor[1]:green value FrontColor[2]:blue value FrontColor[3]: alpha value
<code>headCover.Rect[0]</code> <code>headCover.Rect[1]</code> <code>headCover.Rect[2]</code> <code>headCover.Rect[3]</code>	integer	Range is [0—8191]. Rect[0]: top left corner x coordinate (left) Rect[1]: top left corner y coordinate (top) Rect[2]: bottom right x coordinate (right) Rect[3]: bottom right y coordinate (bottom)
<code>headChannelTitle.BackColor[0]</code> <code>headChannelTitle.BackColor[1]</code> <code>headChannelTitle.BackColor[2]</code> <code>headChannelTitle.BackColor[3]</code>	integer	Range is the same with <code>headCover</code> .
<code>headChannelTitle.EncodeBlend</code>	bool	
<code>headChannelTitle.FrontColor[0]</code> <code>headChannelTitle.FrontColor[1]</code> <code>headChannelTitle.FrontColor[2]</code> <code>headChannelTitle.FrontColor[3]</code>	integer	
<code>headChannelTitle.Rect[0]</code> <code>headChannelTitle.Rect[1]</code> <code>headChannelTitle.Rect[2]</code> <code>headChannelTitle.Rect[3]</code>	integer	
<code>headUserDefinedTitle.Text</code>	string	Title content. Character '\n' means newlines. Limit: Support 2 lines at most.
<code>headUserDefinedTitle.TextAlign</code>	integer	Range is the same with <code>headChannelTitle</code> These are configs about User-defined title.
<code>headUserDefinedTitle.BackColor[0]</code> <code>headUserDefinedTitle.BackColor[1]</code> <code>headUserDefinedTitle.BackColor[2]</code> <code>headUserDefinedTitle.BackColor[3]</code>	integer	
<code>headUserDefinedTitle.EncodeBlend</code>	bool	
<code>headUserDefinedTitle.FrontColor[0]</code> <code>headUserDefinedTitle.FrontColor[1]</code> <code>headUserDefinedTitle.FrontColor[2]</code> <code>headUserDefinedTitle.FrontColor[3]</code>	integer	
<code>headUserDefinedTitle.Rect[0]</code>	integer	

ParamName	ParamValue type	Description
<code>headUserDefinedTitle.Rect[1]</code> <code>headUserDefinedTitle.Rect[2]</code> <code>headUserDefinedTitle.Rect[3]</code>		
<code>headTimeTitle.BackColor[0]</code> <code>headTimeTitle.BackColor[1]</code> <code>headTimeTitle.BackColor[2]</code> <code>headTimeTitle.BackColor[3]</code>	integer	
<code>headTimeTitle.EncodeBlend</code>	bool	
<code>headTimeTitle.FrontColor[0]</code> <code>headTimeTitle.FrontColor[1]</code> <code>headTimeTitle.FrontColor[2]</code> <code>headTimeTitle.FrontColor[3]</code>	integer	Range is the same with <code>headChannelTitle</code> . These are configs about time title.
<code>headTimeTitle.Rect[0]</code> <code>headTimeTitle.Rect[1]</code> <code>headTimeTitle.Rect[2]</code> <code>headTimeTitle.Rect[3]</code>	integer	
<code>headTimeTitle.ShowWeek</code>	bool	True: Display week within the time title.

4.5.12 Get Video Input Capability

Table 4-12

Syntax	<code>http://<server>/cgi-bin/devVideoInput.cgi?action=getCaps&channel=<ChannelNo></code>
Method	GET
Description	Get video input capabilities.
Example	<code>http://192.168.1.250/cgi-bin/devVideoInput.cgi?action=getCaps&channel=1</code>
Success Return	<pre> caps.AutoSyncPhase=false caps.Backlight=2 caps.BrightnessCompensation=true caps.ChipID=0 caps.CoverCount=4 caps.CoverType=1 caps.CustomManualExposure=true caps.DayNightColor=true caps.DayNightColorIO=false caps.DoubleExposure=0 caps.DownScaling=false caps.EEModeRange=100 caps.ElectricFocus=false caps.Exposure=16 caps.ExposureMode=31 caps.ExternalSyncInput=false caps.FishEye=false caps.FlashAdjust=false caps.Flip=true caps.FormatCount=5 </pre>

```
caps.Gain=true
caps.GainAuto=true
caps.Gamma=true
caps.GammaModeRange=100
caps.GlareInhibition=1
caps.HorizontalBinning=0
caps.IRCUT=true
caps.ImageEnhancement.LevelRange[0]=0
caps.ImageEnhancement.LevelRange[1]=100
caps.ImageEnhancement.Support=true
caps.InfraRed=true
caps.Iris=true
caps.IrisAuto=true
caps.LadenBitrate=972000
caps.LimitedAutoExposure=true
caps.MaxExposureTime=300.0
caps.MaxExposureTime1=0.0
caps.MaxHeight=1080
caps.MaxMultiProfile=3
caps.MaxWidth=1920
caps.MeteringRegionCount=0
caps.MinExposureTime=1.0
caps.MinExposureTime1=0.0
caps.Mirror=true
caps.MultiOptions=false
caps.NightOptions=true
caps.ReferenceLevel=false
caps.Rotate90=true
caps.SetColor=true
caps.SignalFormats=Inside
caps.SignalType[0]=VGA
caps.SnapshotExposure=false
caps.SupportProfile=false
caps.SupportWhiteLevel=true
caps.SupportWriteLevel=false
caps.SyncChipChannels=false
caps.SyncFocus=false
caps.TitleCount=4
caps.TridimDenoise=2
caps.TridimDenoiseDetails=0
caps.UTC=0
caps.UpScaling=false
caps.Version=0
caps.VerticalBinning=0
caps.VideoInDenoise.2D.LevelRange[0]=0
caps.VideoInDenoise.2D.LevelRange[1]=100
caps.VideoInDenoise.2D.Support=true
```

	<p>caps.VideoInDenoise.3D.3DAutoType.ModRange[0]=0 caps.VideoInDenoise.3D.3DAutoType.ModRange[1]=100 caps.VideoInDenoise.3D.Support=true caps.VideoInDenoise.Support=true caps.WhiteBalance=3 caps.WideDynamicRange=1</p>
Comment	<p>Parameters in URL: ChannelNo: integer, video channel index which starts from 1. Parameters in Response: see table below</p>

Appendix:

Field in response	Value type	Description
Backlight	integer	0: not support backlight ; 1: support backlight; 2: support regional regulation
ChipID	String	ID of chips in this channel
CoverCount	integer	The maximum cover region count.
CoverType	integer	0: don't support cover 1: support realtime cover 2: support non-realtime cover
CustomManualExposure	bool,	true: support use defined manual exposure time
DayNightColor	bool	true: support color alternate between day and night.
DownScaling	bool	true: support down scaling, binning mode not included.
Exposure	integer	Exposure grade. 0 — don't support exposure control.
ExternalSyncInput	bool	true: support HD signal external synchronization.
FlashAdjust	bool	true: support flash adjust
Flip	bool	true: support picture flip.
Gain	bool	true: support gain control.
GainAuto	bool	true: support auto gain.
HorizontalBinning	integer	Horizontal/Vertical pixel binning mask, 1 — support 2 pixel binning, 2 — support 3 pixel binning 4 — support 4 pixel binning ... 2^n — support $n+2$ pixel binning
VerticalBinning	integer	
InfraRed	bool	true: support Infra compensation
Iris	bool	true: support Iris adjust
IrisAuto	bool	true: support auto Iris adjust
LadenBitrate	integer	Unit is Kbps. Maximum value of video stream bitrates, 16bpp, not in binning mode.
LimitedAutoExposure	bool	true: support auto exposure with time limit.
MaxExposureTime	double	upper limit of exposure time, unit: ms
MaxExposureTime1	double	upper limit of exposure time, unit: ms

Field in response	Value type	Description
MaxHeight	integer	Maximum video height
MaxWidth	integer	Maximum video width
MinExposureTime	double	lower limit of exposure time, unit: ms
MinExposureTime1	double	lower limit of exposure time, unit: ms
Mirror	bool	true: support picture mirror.
NightOptions	bool	true: support night options.
ReferenceLevel	bool	true: support reference level.
Rotate90	bool	true: support clockwise/anticlockwise 90°rotate
SetColor	bool	true: support color set.
SignalFormats	string	<p>It's a string contains supported video input signal formats for this channel. Signal formats are separated by comma. Range is {Inside, BT656, 720p, 1080p, 1080i, 1080sF, 1_3M}</p> <p>Inside — inside input. 1_3M — 1280*960</p>
SyncChipChannels	bool	True: channels in same chip should be synchronized. Synchronized means video resolution of these channels should be the same.
TitleCount	integer	Maximum count of blending titles.
UpScaling	bool	true: support up scaling.
WhiteBalance	integer	<p>Range is {0, 1, 2, 3}</p> <p>0: don't support white balance.</p> <p>1: support auto white balance</p> <p>2: support auto and pre defined white balance.</p> <p>3: support auto, pre defined and user defined white balance</p>

4.5.13 Get Coordinates of Current Window

Table 4-13

Syntax	http://<server>/cgi-bin/devVideoInput.cgi?action=getCurrentWindow&channel=< ChannelNo >
Method	GET
Description	Get the coordinates of the current window.
Example	http://192.168.1.250/cgi-bin/devVideoInput.cgi?action=getCurrentWindow&channel=1
Success Return	<pre>rect[0]=500 rect[1]=500 rect[2]=5000 rect[3]=5000</pre>
Comment	<p>Parameters in URL:</p> <p>ChannelNo: integer, video channel index which starts from 1.</p> <p>Parameters in Response:</p> <p>rect[n] : relative coordinates, range is 0-8192.{0,0,0,0} top-left,</p>

4.5.14 Set Coordinates of Current Window

Table 4-14

Syntax	<code>http://<server>/cgi-bin/devVideoInput.cgi?action=setCurrentWindow&channel=<ChannelNo>&rect[0]=<rect0>&rect[1]=<rect1>&rect[2]=<rect2>&rect[3]=<rect3></code>
Method	GET
Description	Set the coordinates of the current window.
Example	<code>http://192.168.1.250/cgi-bin/devVideoInput.cgi?action=setCurrentWindow&channel=1&rect[0]=0&rect[1]=0&rect[2]=5000&rect[3]=5000</code>
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>ChannelNo: integer, video channel index which starts from 1.</p> <p>rect0 & rect1 & rect2 & rect3: relative coordinates, range is 0-8192.{0,0,0,0} top-left, {8192,0,0,0} top-right, {0,8192,0,0} bottom-left, {8192,8192,0,0} bottom-right</p>

4.5.15 [Config] Video Out

- Get video out config

Table 4-15

Syntax	<code>http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=VideoOut</code>
Method	GET
Description	Get video out config.
Example	<code>http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=VideoOut</code>
Success Return	<p>head.Margin[0]=0 head.Margin[1]=0 head.Margin[2]=0 head.Margin[3]=0 head.Color.Brightness=50 head.Color. Contrast=50 head.Color. Saturation=50 head.Color. Hue=50 head.Mode. Width=800 head.Mode. Height=600 head.Mode. BPP=16 head.Mode. Format="Auto" head.Mode. RefreshRate=60 ...</p>
Comment	<p>Parameters in Response:</p> <p>head = table.VideoOut[ChannelNo].</p> <p>ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p>

- Set video out config

Table 4-16

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set video out config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&table.VideoOut[1].Color.Brightness=50
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>In table below,</p> <p>head = table.VideoOut[ChannelNo].</p> <p>ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p>

Appendix:

ParamName	ParamValue type	Description
head . Margin[0]		
head . Margin[1]	integer	Margin.
head . Margin[2]		
head . Margin[3]		
head . Color.Brightness	integer	Brightness.
head . Color.Contrast =50	integer	Contrast.
head . Color.Satuation =50	integer	Saturation.
head . Color.Hue =50	integer	Hue.
head . Mode.Width =800	integer	Resolution.
head . Mode.Height=600		
head . Mode.BPP =16	integer	Bits per pixel.
head .Mode.Format="Auto"	string	The range is {"Auto","TV","VGA","DVI","HDMI","SPOT","SDI","HDMI/VGA"}.
head .Mode.RefreshRate=60	integer	Refresh rate.

4.5.16 [Config] Smart Encode

- Get Smart Encode config

Table 4-17

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=SmartEncode
Method	GET
Description	Get smart encode config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=SmartEncode
Success Return	head.Enable=false head.Extra[0]=false

Comment	Parameters in Response: head = table.SmartEncode[ChannelNo]. ChannelNo : integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).
---------	---

- Set Smart Encode config

Table 4-18

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&< paramName >=< paramValue >[&< paramName >=< paramValue >...]
Method	GET
Description	Set smart encode config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&SmartEncode[0].Enable=true
Success Return	OK
Comment	Parameters in URL: In table below, head = SmartEncode[ChannelNo]. ChannelNo : integer, array index starts from 0, which means video channel>equals to video channel index -1, and so 0 means channel 1).

Appendix:

ParamName	ParamValue type	Description
head .Enbale	bool	Open or close the smart encode of the main stream.
head .Extra[0]	bool	Open or close the smart encode of the extra stream.

4.5.17 Get Decoder Caps

URL	http://<server>/cgi-bin/DevVideoDec.cgi?action=getCaps					
Method	GET					
Description	Get capability of decoder					
[Request Params] (None)						
[Response Params] (key=value format)						
Name	Type	R/O	Param Description			
Video	object	R	The video caps			
+StreamType	array<string>	R	The stream types that the device supports, ex : h264, h265 etc.			
+LadenBitrate	int	R	The maximum decoding capability that the device supports (w * h * FPS * 16)/1024, uint: kbps, ex : 7680*4320*30*8/1024 = 7776000			
[Example]						
Request	GET http://192.168.1.250/cgi-bin/DevVideoDec.cgi?action=getCaps					
Response	Video.StreamType[0]=h264 Video.StreamType[1]=h265 Video.LadenBitrate=7776000					

4.5.18 [Config] PrivacyMasking

- Get privacy masking config

Syntax	<a href="http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=PrivacyMasking">http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=PrivacyMasking					
Method	GET					
Description	Get privacy masking setting					
[Request Params] (None)						
[Response Params] (key=value format)						
Name	Type	R/O	Param Description			
table	object	R	Config info table.			
+PrivacyMasking	Array<Array<object>>	R	PrivacyMasking config, each channel has several masking region			
++Enable	bool	R	Enable/Disable			
++Postion	Array<double>	O	Size is 3 Postion[0] is horizontal angle, range is [-1—1] Postion[1] is vertical angel, range is [-1—1] Postion[2] is zoom, range is [0—1]			
++ShapeType	string	O	Shape type, can be: "Rect", "Polygon"			
++Rect	Array<int>	O	It is valid if ShapeType is Rect , top left and bottom right point, each point has x and y value, coordinate remap to 0 — 8192.			
++Polygon	Array<Array<int>>	O	It is valid if ShapeType is Polygon the first array is point list, minimum item is 3, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.			
++Color	Array<int>	O	Mask color, RGBA color, 4 integer range is 0-255			
++Mosaic	int	O	Mosaic type, Range is {0, 8, 16, 24, 32} 0: no mosaic 8: mosaic size 8*8 16: mosaic size 16*16 24: mosaic size 24*24 32: mosaic size 32*32			
++ViewAngle	double	O	View angle, range is [0.0, 360.0], unit: degree			
[Example]						
Request	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=PrivacyMasking					

Response	<pre> table.PrivacyMasking[0][0].Enable=true table.PrivacyMasking[0][0].Postion[0]=0.0 table.PrivacyMasking[0][0].Postion[1]=0.0 table.PrivacyMasking[0][0].Postion[2]=1.0 table.PrivacyMasking[0][0].ShapeType=Rect table.PrivacyMasking[0][0].Rect[0]=0 table.PrivacyMasking[0][0].Rect[1]=0 table.PrivacyMasking[0][0].Rect[2]=50 table.PrivacyMasking[0][0].Rect[3]=100 table.PrivacyMasking[0][0].Color[0]=128 table.PrivacyMasking[0][0].Color[1]=128 table.PrivacyMasking[0][0].Color[2]=128 table.PrivacyMasking[0][0].Color[3]=255 table.PrivacyMasking[0][0].Mosaic=8 table.PrivacyMasking[0][0].ViewAngle=30.0 table.PrivacyMasking[0][1].Enable=true ... </pre>
----------	---

- Set privacy masking config

Syntax	<a href="http://<server>/cgi-bin/configManager.cgi?action=setConfig">http://<server>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set privacy masking setting		
[Request Params] (key=value format)			
Name	Type	R/O	Param Description
PrivacyMasking	Array<Array<object>>	R	privacy masking config, each channel has several masking region
+Enable	bool	R	Enable/Disable
+Postion	Array<double>	O	Size is 3 Postion[0] is horizontal angle, range is [-1—1] Postion[1] is vertical angel, range is [-1—1] Postion[2] is zoom, range is [0—1]
+ShapeType	string	O	Shape type,can be: "Rect", "Polygon"
+Rect	Array<int>	O	It is valid if ShapeType is Rect , top left and bottom right point, each point has x and y value, oordinate remap to 0 — 8192.
+Polygon	Array<Array<int>>	O	It is valid if ShapeType is Polygon the first array is point list, minimum item is 3, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.
+Color	Array<int>	O	Mask color, RGBA color, 4 integer range is 0-255
+Mosaic	int	O	Mosaic type, Range is {0, 8, 16, 24, 32} 0: no mosaic 8: mosaic size 8*8 16: mosaic size 16*16 24: mosaic size 24*24 32: mosaic size 32*32
+ViewAngle	double	O	View angle, range is [0.0, 360.0], unit: degree

[Response Params] (OK)	
[Example]	
Request	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&PrivacyMasking[0][0].Enable=true&PrivacyMasking[0][0].Postion[0]=0.0&PrivacyMasking[0][0].Postion[1]=0.0&PrivacyMasking[0][0].Postion[2]=1.0&PrivacyMasking[0][0].ShapeType=Rect&PrivacyMasking[0][0].Rect[0]=0&PrivacyMasking[0][0].Rect[1]=0&PrivacyMasking[0][0].Rect[2]=50&PrivacyMasking[0][0].Rect[3]=100&PrivacyMasking[0][0].Mosaic=8&PrivacyMasking[0][0].ViewAngle=30.0
Response	OK

4.5.19 Get Privacy Masking

Syntax	http://<server>/cgi-bin/PrivacyMasking.cgi?action=getPrivacyMasking		
Method	GET		
Description	Get privacy masking		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
channel	int	R	video channel index which starts from 1
offset	int	R	Offset in the result record set, range is [0, Total – 1].
limit	int	R	Count of result to get.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
total	int	R	Count of result
PrivacyMasking	Array<object >	R	PrivacyMasking config
+Index	int	R	Index of PrivacyMasking which starts from 0
+Name	string	O	PrivacyMasking name
+Enable	int	R	1: enable; 0: disable
+ShapeType	string	O	Shape type, can be: "Rect", "Polygon"
+Rect	Array<int>	O	It is valid if ShapeType is Rect , top left and bottom right point, each point has x and y value, coordinate remap to 0 — 8192.
+Polygon	Array<Array<int>>	O	It is valid if ShapeType is Polygon the first array is point list, minimum item is 3, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.
+Color	Array<int>	O	Mask color, RGBA color, 4 integer range is 0-255
+Mosaic	int	O	Mosaic type, Range is {0, 8, 16, 24, 32} 0: no mosaic 8: mosaic size 8*8 16: mosaic size 16*16 24: mosaic size 24*24 32: mosaic size 32*32
[Example]			
Request	http://192.168.1.250/cgi-bin/PrivacyMasking.cgi?action=getPrivacyMasking&channel=1&offset=2&limit =5		

Response	<pre> total=5 PrivacyMasking[0].Index=0 PrivacyMasking[0].Name=Privacy Masking1 PrivacyMasking[0].Enable=1 PrivacyMasking[0].ShapeType=Polygon PrivacyMasking[0].Polygon[0][0]= 0 PrivacyMasking[0].Polygon[0][1]=0 PrivacyMasking[0].Polygon[1][0]=128 PrivacyMasking[0].Polygon[1][1]=128 PrivacyMasking[0].Polygon[2][0]=256 PrivacyMasking[0].Polygon[2][1]=200 ... PrivacyMasking[0].Color[0]=128, PrivacyMasking[0].Color[1]=128, PrivacyMasking[0].Color[2]=128, PrivacyMasking[0].Color[3]=255, PrivacyMasking[0].Mosaic=8, PrivacyMasking[1].Index =1, PrivacyMasking[1].Enable=1, ... </pre>
----------	--

4.5.20 Set Privacy Masking

Syntax	http://<server>/cgi-bin/PrivacyMasking.cgi?action=setPrivacyMasking		
Method	GET		
Description	Set privacy masking		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
channel	int	R	video channel index which starts from 1
PrivacyMasking	object	R	Privacy masking config
+Index	int	R	Index of PrivacyMasking which starts from 0
+Name	string	O	PrivacyMasking name
+Enable	int	R	1: enable; 0: disable
+ShapeType	string	R	Shape type,can be: "Rect", "Polygon"
+Rect	Array<int>	O	It is valid if ShapeType is Rect , top left and bottom right point, each point has x and y value, coordinate remap to 0 — 8192. mandatory when ShapeType is "Rect"
+Polygon	Array<Array<int>>	O	It is valid if ShapeType is Polygon the first array is point list, minimum item is 3, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192. mandatory when ShapeType is "Polygon"
+Color	Array<int>	O	Mask color, RGBA color, 4 integer range is 0-255. mandatory when Mosaic is 0

+Mosaic	int	R	Mosaic type, Range is {0, 8, 16, 24, 32} 0: no mosaic 8: mosaic size 8*8 16: mosaic size 16*16 24: mosaic size 24*24 32: mosaic size 32*32
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/PrivacyMasking.cgi?action=setPrivacyMasking&channel=2&PrivacyMasking.Index=3&PrivacyMasking.Name=Privacy%20Masking1&PrivacyMasking.Enable=1&PrivacyMasking.ShapeType=Rect&PrivacyMasking.Rect[0]=0&PrivacyMasking.Rect[1]=0&PrivacyMasking.Rect[2]=50&PrivacyMasking.Rect[3]=100&PrivacyMasking.Mosaic=8		
Response	OK		

4.5.21 Goto Privacy Masking

Syntax	http://<server>/cgi-bin/PrivacyMasking.cgi?action=gotoPrivacyMasking		
Method	GET		
Description	Goto privacy masking		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
channel	int	R	video channel index which starts from 1
index	int	R	Index of PrivacyMasking which starts from 0
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/PrivacyMasking.cgi?action=gotoPrivacyMasking&channel=2&index=3		
Response	OK		

4.5.22 Delete Privacy Masking

Syntax	http://<server>/cgi-bin/PrivacyMasking.cgi?action=deletePrivacyMasking		
Method	GET		
Description	Delete privacy masking		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
channel	int	R	video channel index which starts from 1
index	int	R	Index of PrivacyMasking which starts from 0
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/PrivacyMasking.cgi?action=deletePrivacyMasking&channel=2&index=3		
Response	OK		

4.5.23 Clear Privacy Masking

Syntax	http://<server>/cgi-bin/PrivacyMasking.cgi?action=clearPrivacyMasking		
Method	GET		
Description	Clear privacy masking		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
channel	int	R	video channel index which starts from 1
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/PrivacyMasking.cgi?action=clearPrivacyMasking&channel=2		
Response	OK		

4.5.24 Get Privacy Masking Rect

Syntax	http://<server>/cgi-bin/PrivacyMasking.cgi?action=getRealRect		
Method	GET		
Description	Get privacy masking rect		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
channel	int	R	video channel index which starts from 1
index	int	R	Index of PrivacyMasking which starts from 0
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
Rect	Array<int>	R	Rect, top left and bottom right point, each point has x and y value, oordinate remap to 0 — 8192.
[Example]			
Request	http://192.168.1.250/cgi-bin/PrivacyMasking.cgi?action=getRealRect&channel=2&index=1		
Response	Rect[0]=0 Rect[1]=0 Rect[2]=50 Rect[3]=100		

4.5.25 [Config] Motion Detection Settings

- Get motion detect config

Table 4-19

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=MotionDetect		
Method	GET		
Description	Motion detect config of a video channel contains Enable, MotionDetectWindow and EventHandler.		
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=MotionDetect		
Success	table.MotionDetect[0].Enable=false		
Return	table.MotionDetect[0].EventHandler.AlarmOut=1		

```
table.MotionDetect[0].EventHandler.AlarmOutChannels[0]=0
table.MotionDetect[0].EventHandler.AlarmOutEnable=true
table.MotionDetect[0].EventHandler.AlarmOutLatch=10
table.MotionDetect[0].EventHandler.BeepEnable=false
table.MotionDetect[0].EventHandler.Dejitter=5
table.MotionDetect[0].EventHandler.Delay=0
table.MotionDetect[0].EventHandler.ExAlarmOut=1
table.MotionDetect[0].EventHandler.ExAlarmOutChannels[0]=0
table.MotionDetect[0].EventHandler.ExAlarmOutEnable=false
table.MotionDetect[0].EventHandler.FlashEnable=false
table.MotionDetect[0].EventHandler.FlashLatch=10
table.MotionDetect[0].EventHandler.LogEnable=true
table.MotionDetect[0].EventHandler.MailEnable=false
table.MotionDetect[0].EventHandler.Matrix=1
table.MotionDetect[0].EventHandler.MatrixChannels[0]=0
table.MotionDetect[0].EventHandler.MatrixEnable=false
table.MotionDetect[0].EventHandler.MessageEnable=false
table.MotionDetect[0].EventHandler.PtzLink[0][0]=None
table.MotionDetect[0].EventHandler.PtzLink[0][1]=1
table.MotionDetect[0].EventHandler.PtzLinkEnable=false
table.MotionDetect[0].EventHandler.Record=1
table.MotionDetect[0].EventHandler.RecordChannels[0]=0
table.MotionDetect[0].EventHandler.RecordEnable=true
table.MotionDetect[0].EventHandler.RecordLatch=10
table.MotionDetect[0].EventHandler.Snapshot=1
table.MotionDetect[0].EventHandler.SnapshotChannels[0]=0
table.MotionDetect[0].EventHandler.SnapshotEnable=false
table.MotionDetect[0].EventHandler.TimeSection[0][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[0][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[0][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[0][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[0][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[0][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[1][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[1][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[1][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[1][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[1][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[1][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[2][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[2][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[2][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[2][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[2][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[2][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[3][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[3][1]=0 00:00:00-23:59:59
```

```
table.MotionDetect[0].EventHandler.TimeSection[3][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[3][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[3][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[3][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[4][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[4][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[4][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[4][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[4][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[4][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[5][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[5][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[5][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[5][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[5][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[5][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[6][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[6][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[6][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[6][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[6][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[6][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TipEnable=false
table.MotionDetect[0].EventHandler.Tour=1
table.MotionDetect[0].EventHandler.TourChannels[0]=0
table.MotionDetect[0].EventHandler.TourEnable=false
table.MotionDetect[0].EventHandler.Voice.AudioFileName=
table.MotionDetect[0].EventHandler.VoiceEnable=false
table.MotionDetect[0].MotionDetectWindow[0].Id=0
table.MotionDetect[0].MotionDetectWindow[0].Name=Region1
table.MotionDetect[0].MotionDetectWindow[0].Region[0]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[1]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[2]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[3]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[4]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[5]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[6]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[7]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[8]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[9]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[10]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[11]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[12]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[13]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[14]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[15]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[16]=4194303
```

```
table.MotionDetect[0].MotionDetectWindow[0].Region[17]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Sensitive=60
table.MotionDetect[0].MotionDetectWindow[0].Threshold=5
table.MotionDetect[0].MotionDetectWindow[0].Window[0]=0
table.MotionDetect[0].MotionDetectWindow[0].Window[1]=0
table.MotionDetect[0].MotionDetectWindow[0].Window[2]=8191
table.MotionDetect[0].MotionDetectWindow[0].Window[3]=8191
table.MotionDetect[0].MotionDetectWindow[1].Id=1
table.MotionDetect[0].MotionDetectWindow[1].Name=Region2
table.MotionDetect[0].MotionDetectWindow[1].Region[0]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[1]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[2]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[3]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[4]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[5]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[6]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[7]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[8]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[9]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[10]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[11]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[12]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[13]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[14]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[15]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[16]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[17]=0
table.MotionDetect[0].MotionDetectWindow[1].Sensitive=60
table.MotionDetect[0].MotionDetectWindow[1].Threshold=5
table.MotionDetect[0].MotionDetectWindow[1].Window[0]=0
table.MotionDetect[0].MotionDetectWindow[1].Window[1]=0
table.MotionDetect[0].MotionDetectWindow[1].Window[2]=0
table.MotionDetect[0].MotionDetectWindow[1].Window[3]=0
table.MotionDetect[0].MotionDetectWindow[2].Id=2
table.MotionDetect[0].MotionDetectWindow[2].Name=Region3
table.MotionDetect[0].MotionDetectWindow[2].Region[0]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[1]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[2]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[3]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[4]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[5]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[6]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[7]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[8]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[9]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[10]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[11]=0
```

```
table.MotionDetect[0].MotionDetectWindow[2].Region[12]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[13]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[14]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[15]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[16]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[17]=0
table.MotionDetect[0].MotionDetectWindow[2].Sensitive=60
table.MotionDetect[0].MotionDetectWindow[2].Threshold=5
table.MotionDetect[0].MotionDetectWindow[2].Window[0]=0
table.MotionDetect[0].MotionDetectWindow[2].Window[1]=0
table.MotionDetect[0].MotionDetectWindow[2].Window[2]=0
table.MotionDetect[0].MotionDetectWindow[2].Window[3]=0
table.MotionDetect[0].MotionDetectWindow[3].Id=3
table.MotionDetect[0].MotionDetectWindow[3].Name=Region4
table.MotionDetect[0].MotionDetectWindow[3].Region[0]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[1]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[2]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[3]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[4]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[5]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[6]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[7]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[8]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[9]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[10]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[11]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[12]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[13]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[14]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[15]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[16]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[17]=0
table.MotionDetect[0].MotionDetectWindow[3].Sensitive=60
table.MotionDetect[0].MotionDetectWindow[3].Threshold=5
table.MotionDetect[0].MotionDetectWindow[3].Window[0]=0
table.MotionDetect[0].MotionDetectWindow[3].Window[1]=0
table.MotionDetect[0].MotionDetectWindow[3].Window[2]=0
table.MotionDetect[0].MotionDetectWindow[3].Window[3]=0
table.MotionDetect[0].OsdTwinkleEnable=false
table.MotionDetect[0].PirMotionLevel=3
```

- Set motion detect config

Table 4-20

Syntax `http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]`

Method	GET
Description	Set motion detection config.
Example	<p>Enable motion detection: http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&MotionDetect[0].Enable=true</p> <p>Set motion detection regions: http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&MotionDetect[0].MotionDetectWindow[0].Region[0]=1&MotionDetect[0].MotionDetectWindow[0].Region[1]=1&MotionDetect[0].MotionDetectWindow[0].Region[2]=1&MotionDetect[0].MotionDetectWindow[0].Region[3]=1&MotionDetect[0].MotionDetectWindow[0].Region[4]=1&MotionDetect[0].DetectVersion=V3.0</p>
Success Return	OK
Comment	<p>In table below, head = MotionDetect[ChannelNo] ChannelNo: integer, array index starts from 0, which means video channel>equals to video channel index -1, and so 0 means channel 1).</p> <p>LineNum Index of region, region is divided into lines and each line has several blocks, a line is described by a 32 bit integer, a bit for a block. 0=Line 1 1=Line 2 ... WinNum Index of detect window, there are 4 detect windows at present. Each window is divided into 18 lines and 22 blocks per line.</p> <p>Notice: When setting "MotionDetect[ChannelNo].MotionDetectWindow[WinNum].Region", you need to contain the parameter "MotionDetect[ChannelNo].DetectVersion=V3.0" along.</p>

Appendix:

ParamName	ParamValue type	Description
head .Enable	bool	Enable/Disable motion detect feature in a channel.
head .PtzManualEnable	bool	Enable/Disable manual control Ptz, only for Hahua devices
head .OsdTwinkleEnable	bool	Osd Twinkle Enable default : false
head .Row	integer	motion detect row number
head .Col	integer	motion detect colum number
head .Level	integer	sensitive ,range 1-6
head .TimeDivide.TimeSchedule	array<array<string>>	TimeSchedule
head .TimeDivide.Level	integer	sensitive in timesection, different with global sensitive.
head .DetectVersion	string	detect version

ParamName	ParamValue type	Description
head.PirMotionEnable	bool	Customized Pir Motion Enable
head.PirMotionLevel	integer	Customized Pir Motion sensitive level
head.VolumeRatio	integer	Area occupancy ratio
head.SubRatio	integer	SubRatio
head.Region	array<integer>	Region
head.EventHandler	—	Setting of EventHandler is described in SetEventHandler .
head.MotionDetectWin.dow [WinNum].Id	integer	It is the Id of a detect window.
head.MotionDetectWin.dow [WinNum].Name	string	It is the name of a detect window.
head.MotionDetectWin.dow[WinNum].Sensitive	integer	Range is [0—100]. Sensitivity of motion detection. It presents more sensitive if the value is larger.
head.MotionDetectWin.dow[WinNum].Threshold	integer	Range is [0—100]. It presents the threshold value when trigger motion detect.
head.MotionDetectWin.dow[WinNum].Region[LineNum]	integer	It is similar with head.Region [LineNum]. Currently, a region is divided into 18 lines and 22 blocks per line. A bit describes a block in the line. Bit = 1: motion in this block is monitored. Example: MotionDetect [0].Region [0] = 4194303 (0x3FFFFF): the 22 blocks in channel 0 line 0 is monitored. MotionDetect [0].Region [1] = 0: the 22 blocks in channel 0 line 1 is not monitored. MotionDetect [0].Region [17] = 3: the left two blocks in the last line of channel 0 is monitored.

4.5.26 [Config] LAEConfig

- Get LAEConfig setting

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=LAEConfig
Method	GET
Description	Get LAE Configs.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=LAEConfig
Success Return	table.LAEConfig.Enable=true table.LAEConfig.ManulEn=1 table.LAEConfig.YSIGMA =1 table.LAEConfig.Dethr =1 table.LAEConfig.STS =1 table.LAEConfig.GauFilter0=1 table.LAEConfig.GauFilter1=1

	table.LAEConfig.GauFilter2=1 table.LAEConfig.UVGauFilter0=1 table.LAEConfig.UVGauFilter1=1 table.LAEConfig.UVGauFilter2=1 table.LAEConfig.UVSigma =1 table.LAEConfig.EEGauFilter0=1 table.LAEConfig.EEGauFilter1=1 table.LAEConfig.EEGauFilter2=1 table.LAEConfig.EEratio =1 table.LAEConfig.GEnable =1 table.LAEConfig.GainLevel =1 table.LAEConfig.DebugPin =1
Comment	

- Set LAEConfig setting

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set LAE Configs.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&LAEConfig.Enable=true&LAEConfig.ManulEn=1
Success Return	OK
Comment	

Appendix:

ParamName	ParamValue type	Description
table.LAEConfig.Enable	bool	turn on/off LAE fuction The following parameters will take effects when it's on
table.LAEConfig.ManulEn	integer	whether use manual mode, 1 for manual mode, 0 for alternative plan
table.LAEConfig.YSIGMA	integer	absolute strength of Y channel Grinding, The smaller the stronger(fuzzier)
table.LAEConfig.Dethr	integer	relative strength of Y channel Grinding, The smaller the stronger(fuzzier)
table.LAEConfig.STS	integer	Y channel STS
table.LAEConfig.GauFilter0	integer	Y channel pre filter0, meet the conditions: GauFilter0 * 2 + GauFilter1 * 2 + GauFilter2 = 256
table.LAEConfig.GauFilter1	integer	Y channel pre filter1
table.LAEConfig.GauFilter2	integer	Y channel pre filter2
table.LAEConfig.UVGauFilter0	integer	UV channel pre filter0, meet the conditions: GauFilter0 * 2 + GauFilter1 * 2 + GauFilter2 = 256
table.LAEConfig.UVGauFilter1	integer	UV channel pre filter1
table.LAEConfig.UVGauFilter2	integer	UV channel pre filter2

ParamName	ParamValue type	Description
table.LAEConfig.UVSigma	integer	absolute strength of UV channel Grinding, The smaller the stronger
table.LAEConfig.EEGauFilter0	integer	EE pre filter0, meet the conditions: GauFilter0 * 2 + GauFilter1 * 2 + GauFilter2 = 256
table.LAEConfig.EEGauFilter1	integer	EE pre filter1, extracts low-frequenc
table.LAEConfig.EEGauFilter2	integer	EE pre filter2
table.LAEConfig.EEratio	integer	strength of enhancement, the bigger the stronger
table.LAEConfig.GEnable	integer	Brightening switch, default false
table.LAEConfig.GainLevel	integer	it is used to choose paramas, as an alternative
table.LAEConfig.DebugPin	integer	debug option

4.5.27 Enable/Disable all privacy masking covers

Syntax	http://<server>/cgi-bin/PrivacyMasking.cgi?action=setPrivacyMaskingEnable		
Method	GET		
Description	Enable/Disable all privacy masking covers		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
channel	int	R	video channel index which starts from 1
Enable	bool	R	true: Enable, false: Disable
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/PrivacyMasking.cgi?action=setPrivacyMaskingEnable&channel=2&Enable=true		
Response	OK		

4.5.28 Get enable/disable state of all privacy masking covers

Syntax	http://<server>/cgi-bin/PrivacyMasking.cgi?action=getPrivacyMaskingEnable		
Method	GET		
Description	Get enable/disable state of all privacy masking covers		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
channel	int	R	video channel index which starts from 1
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
Enable	bool	R	true: Enable, false: Disable
[Example]			
Request	http://192.168.1.250/cgi-bin/PrivacyMasking.cgi?action=getPrivacyMaskingEnable&channel=2		
Response	Enable=false		

4.6 System

4.6.1 [Config] General

- Get general config

Table 4-21

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=General
Method	GET
Description	Get general config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=General
Success Return	table.General.LocalNo=8, table.General.MachineID=20832748927, table.General.MachineName=Device001, table.General.MachineAddress=XXXDistrict.YYYRoad table.General.MachineGroup=ZZZGroup table.General.LockLoginEnable=true, table.General.CheckDuration=30, table.General.LockLoginTimes=3, table.General.LoginFailLockTime=1800, table.General.MaxOnlineTime=1800, table.General.LocalPolicy.LockLoginEnable=true, table.General.LocalPolicy.CheckDuration=30, table.General.LocalPolicy.LockLoginTimes=3, table.General.LocalPolicy.LoginFailLockTime=1800 table.General.ActivationTime=2000-01-01 00:00:00
Comment	—

- Set general config

Table 4-22

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue> [&<paramName>=<paramValue> ...]
Method	GET
Description	Set general config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&General.MachineName=MyIPC
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
General.MachineName	string	Device name or serial number.
General.LocalNo	integer	Local number for remote controller

ParamName	ParamValue type	Description
General.MachineAddress	string	Address machine places in
General.MachineGroup	string	Group machine belongs to

4.6.2 Get Current Time

Table 4-23

Syntax	http://<server>/cgi-bin/global.cgi?action=getCurrentTime
Method	GET
Description	Get current time.
Example	http://192.168.1.250/cgi-bin/global.cgi?action=getCurrentTime
Success Return	result = 2011-7-3 21:02:32
Comment	The time format is "Y-M-D H-m-S". It's not be effected by Locales. TimeFormat in SetLocalesConfig.

4.6.3 Set Current Time

Table 4-24

Syntax	http://<server>/cgi-bin/global.cgi?action=setCurrentTime&time=2011-7-3%2021:02:32
Method	GET
Description	Set current time.
Example	http://192.168.1.250/cgi-bin/global.cgi?action=setCurrentTime&time=2016-01-01%2021:02:32
Success Return	OK
Comment	The time format is "Y-M-D H-m-S". It's not be effected by Locales. TimeFormat in SetLocalesConfig.

4.6.4 [Config] Locales

- Get locales config

Table 4-25

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=Locales
Method	GET
Description	Get locales config
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=Locales
Success Return	table.Locales.DSTEnable=false table.Locales.DSTEEnd.Day=1 table.Locales.DSTEEnd.Hour=0 table.Locales.DSTEEnd.Minute=0 table.Locales.DSTEEnd.Month=1 table.Locales.DSTEEnd.Week=2

	table.Locales.DSTEnd.Year=2011 table.Locales.DSTStart.Day=0 table.Locales.DSTStart.Hour=0 table.Locales.DSTStart.Minute=0 table.Locales.DSTStart.Month=1 table.Locales.DSTStart.Week=1 table.Locales.DSTStart.Year=2011 table.Locales.TimeFormat=yyyy-MM-dd HH:mm:ss
Comment	—

- Set locales config

Table 4-26

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&< paramName >=< paramValue >[&< paramName >=< paramValue >...]
Method	GET
Description	Set Locales config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&Locales.DSTEnable=false
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
Locales.DSTEnable	bool	Enable/Disable DST (daylight saving time)
Locales.DSTEnd.Day	integer	Range is [0—6] or [1—31] [0—6]: week day, 0 = Sunday, 6 = Saturday [1—31]: month day If Locales.DSTEnd.Week is 0, use month day, otherwise, use week day.
Locales.DSTEnd.Hour	integer	Range is [0—23]
Locales.DSTEnd.Minute	integer	Range is [0—59]
Locales.DSTEnd.Month	integer	Range is [1—12]
Locales.DSTEnd.Week	Integer	Range is {1, 2, 3, 4, -1, 0}. 0 = Use month day [1, 2, 3, 4, -1]: use week day. 1 = first week, 2 = second, 3 = third, 4 = fourth, -1 = last.
Locales.DSTEnd.Year	Integer	Range is [2000-2038]
Locales.DSTStart.Day	Integer	Range is the same with items in Locales.DSTEnd. Locales.DSTStart table and Locales.DSTEnd table define the time range of DST.
Locales.DSTStart.Hour		
Locales.DSTStart.Minute		
Locales.DSTStart.Month		
Locales.DSTStart.Week		
Locales.DSTStart.Year		

ParamName	ParamValue type	Description
Locales.TimeFormat	string	<p>Defines time format displayed in video time title. String form is: <i>year-month-day hour:mm:ss</i>. Position of <i>year</i>, <i>month</i> and <i>day</i> can be exchanged.</p> <p>Range of <i>year</i> is {yy, yyyy} yy = year without century, yyyy = year with century. Range of <i>month</i> is {M, MM, MMMM} M = 1 for January, MM = 01 for January, MMMM = Jan for January Range of <i>day</i> is {d, dd} d = 1 for first day, dd = 01 for first day Range of <i>hour</i> is {H, HH, h, hh} H = 1 for 1:00, HH = 01 for 1:00, range is 0-23 h = 1 for 1:00, hh = 01 for 1:00, time range is 1-12</p> <p>Example: yyyy-MM-dd HH:mm:ss or MM-dd-yyyy HH:mm:ss or dd-M-yy hh:mm:ss</p>

4.6.5 [Config] Holiday Management

- Get holiday config

Table 4-27

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=Holiday
Method	GET
Description	Get holiday config for record or snap.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=Holiday
Success Return	<pre>table.Holiday.MonthMask[0]=3 table.Holiday.MonthMask[1]=0 table.Holiday.MonthMask[2]=0 table.Holiday.MonthMask[3]=0 table.Holiday.MonthMask[4]=0 table.Holiday.MonthMask[5]=0 table.Holiday.MonthMask[6]=0 table.Holiday.MonthMask[7]=0 table.Holiday.MonthMask[8]=0 table.Holiday.MonthMask[9]=1610612739 table.Holiday.MonthMask[10]=0 table.Holiday.MonthMask[11]=0</pre>
Comment	—

- Set holiday config

Table 4-28

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set holiday config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&Holiday.MonthMask[0]=3
Success Return	OK
Comment	Parameters in URL: In table below, monthIndex presents the index of a month. 0 presents January, 1 presents February, 11 presents December.

Appendix:

ParamName	ParamValue type	Description
Holiday.MonthMask[monthIndex]	integer	It is the mask of a month. Every bit present a day. For example, 0x0001 presents the first day of a month is holiday. 0x0002 presents the second day of a month is holiday, 0x0003 presents the first day and second day of a month is holiday.

4.6.6 Get Language Capability

Table 4-29

Syntax	http://<server>/cgi-bin/magicBox.cgi?action=getLanguageCaps
Method	GET
Description	Get the list of supported languages.
Example	http://192.168.1.250/cgi-bin/magicBox.cgi?action=getLanguageCaps
Success Return	Languages=SimpChinese,English,French
Comment	response is a string contains languages with comma separated. Languages include { English, Italian, Spanish, Japanese, Russian, French, German }

4.6.7 [Config] Language

- Get language config

Table 4-30

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=Language
Method	GET
Description	Get system language config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=Language
Success	table.Language=SimpChinese

Return	
Comment	—

- Set language config

Table 4-31

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set system language config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&Language=SimpChinese
Success Return	OK
Comment	NOTE: After changing language setting, system will automatically reboot!

Appendix:

ParamName	ParamValue type	Description
Language	string	The language range is get from interface in GetLanguageCaps

4.6.8 Get Device Type

Get the device type displayed (instead of the real type).

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=getDeviceType					
Method	GET					
Request Params (none)						
Request Example						
http://192.168.1.250/cgi-bin/magicBox.cgi?action=getDeviceType						

Response Params (key=value format in body)				
Name	Type	R/O	Description	Sample
type	string	R	The displayed device model	DVR
Response Example				
type=DVR				

4.6.9 Get Hardware Version

Get the device hardware version information.

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=getHardwareVersion					
Method	GET					
Request Params (none)						
Request Example						
http://192.168.1.250/cgi-bin/magicBox.cgi?action=getHardwareVersion						

Response Params (key=value format in body)				
Name	Type	R/O	Description	Sample
version	string	R	The hardware version in the format of xx.xx. Use two numbers to respectively represent the main and sub versions. If the beginning number of the main version is 0, it should be omitted.	1.00
Response Example				
version=1.00				

4.6.10 Get Serial Number of Device

Get the serial number of the device.

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=getSerialNo					
Method	GET					
Request Params (none)						
Request Example						
http://192.168.1.250/cgi-bin/magicBox.cgi?action=getSerialNo						

Response Params (key=value format in body)				
Name	Type	R/O	Description	Sample
sn	string	R	The device serial number	YZC0GZ05100020
Response Example				
sn=YZC0GZ05100020				

4.6.11 Get Machine Name

Get the device machine name.

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=getMachineName					
Method	GET					
Request Params (none)						
Request Example						
http://192.168.1.250/cgi-bin/magicBox.cgi?action=getMachineName						

Response Params (key=value format in body)				
Name	Type	R/O	Description	Sample
name	string	O	The device name	my machine
Response Example				
name=my machine				

4.6.12 Get System Information1

Get the system information of the device (No longer maintained).

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=getSystemInfo		
Method	GET		

Request Params (none)**Request Example**

```
http://192.168.1.250/cgi-bin/magicBox.cgi?action=getSystemInfo
```

Response Params (key=value format in body)

Name	Type	R/O	Description	Sample
serialNumber	string	O	Device serial number	PA1FQ15900207
deviceType	int	O	Device model	27
processor	string	O	Processor model	ST7108

Response Example

```
serialNumber=PA1FQ15900207
```

```
deviceType=27
```

```
processor=ST7108
```

4.6.13 Get System Information2

Get the system information of the device.

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=getSystemInfoNew
--------------------	--

Method	GET
---------------	-----

Request Params (none)

Request Example

http://192.168.1.250/cgi-bin/magicBox.cgi?action=getSystemInfoNew

Response Params (key=value format in body)

Name	Type	R/O	Description	Sample
info	object	O	System information	
+SyncTime	object	O	For devices without RTC or when the RTC of the device is out of battery, you might also need to synchronize the time.	
++Strategy	enumchar[16]	R	Time synchronization strategy enumchar[16]{ “None” “PerLogin”: Synchronizes the time each time you log in }	PerLogin
+2DCode	char[32]	O	security code 0-9 and capitalized A-Z	123456
+TotalRunTime	uint64	O	The total running time of the device. Unit: s	12456
+cameraNum	uint8	O	The number of cameras	1
+cardReader	bool	O	Whether card swiping is supported: true: supported false: not supported	true
+flashID	uint8[8]	O	flash ID	[200,209,128,149, 64,127,127,200]

+hasRTC	bool	O	Whether the RTC chip is included (for recording the system time). If the file does not exist, the value is true and the RTC is included by default. RTC included: true No RTC: false	true
---------	------	---	---	------

Response Example

```
info.SyncTime.Strategy="PerLogin",
info.2DCode="123456",
info.TotalRunTime=123456
info.cameraNum=0
info.cardReader=false
info.flashID[0]=200
info.flashID[1]=209
info.flashID[2]=128
info.flashID[3]=149
info.flashID[4]=64
info.flashID[5]=127
info.flashID[6]=127
info.flashID[7]=200
info.hasRTC=true
```

4.6.14 Get Vendor Information

Table 4-32

Syntax	http://<server>/cgi-bin/magicBox.cgi?action=getVendor
Method	GET
Description	Get the vendor information.
Example	http://192.168.1.250/cgi-bin/magicBox.cgi?action=getVendor
Success Return	vendor=TTT
Comment	—

4.6.15 Get Software Information

Table 4-33

Syntax	http://<server>/cgi-bin/magicBox.cgi?action=getSoftwareVersion
Method	GET
Description	Get the software information.
Example	http://192.168.1.250/cgi-bin/magicBox.cgi?action=getSoftwareVersion
Success Return	version=2.212.0000.0.R,build:2013-11-14
Comment	—

4.6.16 Get Version of Onvif

Table 4-34

Syntax	http://<server>/cgi-bin/IntervideoManager.cgi?action=getVersion&Name=Onvif
Method	GET
Description	Get onvif version.
Example	http://192.168.1.250/cgi-bin/IntervideoManager.cgi?action=getVersion&Name=Onvif
Success Return	version=2.4.2
Comment	—

4.6.17 Get Version of HTTP API

Table 4-35

Syntax	http://<server>/cgi-bin/IntervideoManager.cgi?action=getVersion&Name=CGI
Method	GET
Description	Get CGI version.
Example	http://192.168.1.250/cgi-bin/IntervideoManager.cgi?action=getVersion&Name=CGI
Success Return	version=2.0.0
Comment	—

4.6.18 Get Device Class

Get the class information of the device.

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=getDeviceClass					
Method	GET					
Request Params (none)						
Request Example						
http://192.168.1.250/cgi-bin/magicBox.cgi?action=getDeviceClass						

Response Params (key=value format in body)				
Name	Type	R/O	Description	Sample
class	string	R	The class information of the device.	HDVR
Response Example				
class=HDVR				

4.6.19 [Config] Auto Maintain

- Get auto maintain config

Table 4-36

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=AutoMaintain
--------	--

Method	GET
Description	Get Auto Maintain config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=AutoMaintain
Success Return	table.AutoMaintain.AutoRebootDay=3 table.AutoMaintain.AutoRebootHour=0 table.AutoMaintain.AutoRebootMinute=0 table.AutoMaintain.AutoShutdownDay=1 table.AutoMaintain.AutoShutdownHour=0 table.AutoMaintain.AutoShutdownMinute=0 table.AutoMaintain.AutoStartUpDay=1 table.AutoMaintain.AutoStartUpHour=2 table.AutoMaintain.AutoStartUpMinute=0

Comment —

- Set auto maintain config

Table 4-37

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set auto maintain config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&AutoMaintain.AutoRebootDay=7
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
AutoMaintain.AutoRebootDay	integer	Range is [-1—7]. Auto restart day. -1 = never auto restart 0—6 = Sunday—Saturday 7 = restart every day
AutoMaintain.AutoRebootHour	integer	Range is [0—23]. Auto restart hour
AutoMaintain.AutoRebootMinute	integer	Range is [0—59]. Auto restart minute
AutoMaintain.AutoShutdownDay	integer	auto reboot time
AutoMaintain.AutoShutdownHour		Range is same with AutoOpenDay, AutoOpenHour and AutoOpenMinute.
AutoMaintain.AutoShutdownMinute		
AutoMaintain.AutoStartUpDay	integer	Auto shutdown time.
AutoMaintain.AutoStartUpHour		Range is same with AutoOpenDay, AutoOpenHour, and AutoOpenMinute.
AutoMaintain.AutoStartUpMinute		

4.6.20 Reboot

Table 4-38

Syntax	http://<server>/cgi-bin/magicBox.cgi?action=reboot[&delay=<paramValue>]
Method	GET
Description	Reboot the device.
Example	http://192.168.1.250/cgi-bin/magicBox.cgi?action=reboot
Success Return	OK
Comment	If successful, response OK. If fail, response Error.

4.6.21 Shutdown

Table 4-39

Syntax	http://<server>/cgi-bin/magicBox.cgi?action=shutdown
Method	GET
Description	Shutdown the device.
Example	http://192.168.1.250/cgi-bin/magicBox.cgi?action=shutdown
Success Return	OK
Comment	If successful, response OK. If fail, response Error.

4.6.22 Factory Reset

Table 4-40

Syntax	http://<server>/cgi-bin/magicBox.cgi?action=resetSystemEx[&type=<type>]
Method	GET
Description	Reset the configuration for the device to the factory default.
Example	http://192.168.1.250/cgi-bin/magicBox.cgi?action=resetSystemEx&type=0
Success Return	OK
Comment	Parameters in URL: Type : integer, it can be 0 or 1. 0 means all parameters are set to their factory default value; 1 means all parameters are set to their factory default value expect the specific parameters; The specific parameters of different device types are different. But it always contains the network settings and user settings. After resetting, the device is reachable on the same ip address as used before the reset. When the Type parameter is not present in the URL, the default value of the Type is 0.

4.6.23 Get Tracing Code of Device

Table 4-41

Syntax	http://<server>/cgi-bin/magicBox.cgi?action=getTracingCode
Method	GET
Description	Get the tracing code of the device.
Example	http://192.168.1.250/cgi-bin/magicBox.cgi?action=getTracingCode
Success Return	tc =0001...1101
Comment	Only system default admin can get it. tc is a 64bit binary string.

4.6.24 Add Camera to Specified Channel

Request URL	http://<server>/cgi-bin/LogicDeviceManager.cgi?action=addCameraByGroup			
Method	POST			
Request Params (Json format in body)				
Parameter	Type	Required	Description	Example
+group	object[]	Yes	Add device groups in batches It is an array with the same elements as addCameraByDevice.	
++DeviceInfo	object	Yes	Device details	
+++RtspPort	int	No	RTSP port: Required when ProtocolType is not Private, 554 by default.	554
+++HttpPort	int	No	HTTP port: Required when ProtocolType is not Private, 80 by default.	80
+++Port	int	No	TCP port: Required when ProtocolType is Private, 37777 by default.	37777
+++UserName	char[32]	Yes	Username	admin
+++Password	char[32]	Yes	Password, plain text	pass123456
+++ProtocolType	char[128]	Yes	Protocol Type "Private","Cpplus2","Cpplus3","General","Onvif","GB28181","Ehome"	"Private"
+++Address	char[16]	Yes	Device address	172.11.1.161
++cameras	object[]	Yes	Video source information	
+++uniqueChannel	int	Yes	Unique channel number, starting from 1	1
Example				
{ "group": [{ "DeviceInfo": { "UserName": "admin", "Password": "pass123456", "ProtocolType": "Private" } }] }				

```

    "ProtocolType": "Private",
    "Port": 37777,
    "Address": "172.11.1.161"
},
"cameras": [
    {
        "uniqueChannel": 1
    }
]
}

```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
group	object[]	O	device group info	
+deviceID	char[128]	O	device ID	"uuid:4848"
+cameras	object[]	O	camera info	
++uniqueChannel	uint	O	unique channel	1
++failedCode	uint	O	error code. 0: succeed 1: the specific channel does not support configuration	1

Response Example

```
{
    "group": [
        {
            "deviceID": "uuid:4848",
            "cameras": [
                {
                    "uniqueChannel": 1,
                    "failedCode": 1
                }, ...
            ]
        }, ...
    ]
}
```

4.6.25 Acquiring All Available Resources

Customer requests to obtain camera information, such as MAC address and SN, through CGI commands.

Request URL	http://<server>/cgi-bin/LogicDeviceManager.cgi?action=getCameraAll			
Method	GET			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Request Example				{}

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
------	------	-----	-------------	---------

camera	object[]	R	camera info	
+ControlID	char[128]	O	<p>Control number.</p> <p>Keyboard users can use the number to locate channels to improve operation efficiency.</p> <p>Control number “0” is not available, which is reserved for special use.</p>	"5001"
+Name	char[128]	O	Name	“XXX Road”
+Enable	bool	O	<p>Enable only when using DevicelID to add/delete devices. Do not enable it through DeviceInfo.</p> <p>true: Meaning trying to connect the device, but not necessarily successful. You can acquire the connection status through getCameraState.</p> <p>false: Do not connect the device. From user's point of view, it is the same as that the device is not added.</p> <p>This Enable field is only used for the case of adding a device through DevicelID. We keep this field for compatibility and recommend using the Enable in DeviceInfo.</p> <p>When acquiring the field, this Enable and the Enable in DeviceInfo are returned at the same time.</p> <p>When sending the field, you can only select one between this Enable and the Enable in DeviceInfo. We recommend sending the Enable in DeviceInfo.</p>	true
+DeviceInfo	DeviceInfo	O	<p>Device details.</p> <p>Use the Enable in the structure to add/delete devices.</p> <p>True: Add devices.</p> <p>False: delete devices from the channel.</p> <p>A device can be added only when "DevicelD" is " " and Info is not NULL.</p>	
++Enable	bool	O	The device is enabled or not.	true
++EncryptStream	int	O	Device stream encryption algorithm (custom field for NVR).	0

			(If other encryption algorithms are used, refer to the encryption type definition in stream format code 0x95). 0: No encryption by default.	
++Address	char[16]	O	Device IP address or domain name.	"10.6.5.10"
++Port	int	O	Port number.	37777
++usePreSecret	enumint8	O	Select whether to use NVR preset password to add cameras (the naming remains the same globally, so the case is not uniform here). enumint8{ 0: Use UserName/Password to connect devices (default). 1: When using a preset password to connect devices, UserName/Password is not required. 2: When using camera login password to connect devices, UserName/Password is not required. }	1
++UserName	char[32]	O	Username	"admin"
++Password	char[32]	O	Password in plain text.	"admin"
++ProtocolType	enumchar[32]]	O	The protocol type of the connected device. For the value range, see the ProtocolType field of the RemoteDevice configuration. enumchar[32]{ "Private" "Cpplus2" "Cpplus3" "General" "Onvif" "Ehome" "ICC": The special protocol for the central Intelligence platform. It has not been reviewed, therefore it is not allowed to add the protocol type in this way in the future. "CpplusEVMS": The special protocol for the central Intelligence platform. It has not been reviewed, therefore it is not allowed to add the protocol type in this way in the	"Private"

			future. }	
++VideoInput++ Channels	uint	O	Total number of video input channels, including analog and digital channels	16
++AudioInput++ Channels	uint	O	Total number of audio input channels	4
++DeviceClass	char[64]	O	Device type	"IPC"
++DeviceType	char[64]	O	Device model	"CP-IPC"
++HttpPort	int	O	HTTP port number	80
++HttpsPort	int	O	HTTPS port number	443
++RtspPort	int	O	RTSP port number	554
++Name	char[64]	O	Device name	"IPC1"
++MachineAddress	char[256]	O	Device deployment location	"XXX District, YYY Road"
++SerialNo	char[48]	O	Device serial number	"Device12345678"
++VendorAbbr	char[32]	O	Vendor abbreviation (optional) Vendor type: VVV (for example).	"VVV"
++HardID	char[64]	O	Hardware ID (optional)	"454"
++SoftwareVersion	char[64]	O	Software version (optional)	"2.420.0000006.R.150311"
++ActivationTime	char[24]	O	The date when starting using the device. Date format: "yyyy-MM-dd HH:mm:ss" "Year–Month–Day Hour: Minute: Second" Range {"2000-01-01 00:00:00", /*Minimum date*/ "2099-01-01 00:00:00", /*Maximum date*/}	"2000-01-01 00:00:00"
++NodeType	char[32]	O	Device node type (main or sub node). It is used when adding storage node (DataNodeManager.addDataNode) in N + M cluster mode.	
++Mac	char[]	O	MAC address	"08:00:20:0A:8C:6D"
++OEMVendor	char[]	O	OEM vendor type. During cloud update, for some vendors, the Vendor field is displayed as OEM, and OEMVendor is added to indicate the actual vendor information. OEMVendor is the same as Vendor by default. If OEMVendor is defined in the product definition,	"XXX"

			refer to the the product definition.	
+DeviceID	char[128]	O	Device ID When DeviceID is not " ", ignore DeviceInfo and use the existing DeviceID information in the RemoteDevice configuration.	"dev123"
+Type	enumchar[32]	O	Input channel type. enumchar[32]{ "Local": Local channel (physical channel). "Remote": Remote channel. "Reserved32": Reserve 32 channels (compatible with the device protocol of the second generation). "Output": Output channel. "Matrix": Analog matrix channel. "Compose": Composite channel. "Cascade": Optical fiber cascading channel. }	"Remote"
+VideoStream	enumchar[32]	O	Video stream. enumchar[32]{ "Main": Main stream "Extra1": Sub stream 1 "Extra2": Sub stream 2 "Extra3": Sub stream 3 "Auto": Automatically selects the appropriate stream. "Preview": Preview raw data stream. "None": No video stream (pure audio stream). }	"Main"
+Channel	uint	O	Channel number of the remote device (optional). DeviceID is unique.	0
+UniqueChannel	int	O	Unique channel number of uniform device number (required). You can obtain video/audio streams by specifying the channel number with 0x11/0xF4 second-generation protocol. Notes: When implementing any addressing function of DeviceID+Channel, LogicDeviceManager.getCameraA	0

			<p>II() must be implemented, and the device must support "DeviceID=Unique" unified numbering method for input channels.</p> <p>For setCamera(), uniqueChannel equals -1 means that the device automatically assigns a channel number.</p>	
+DataNodeIdx	int	O	<p>Indicates the data node to which the global channel is currently allocated (N+0 cluster).</p> <p>Counted from 0. If it is -1 or when the field does not exist, it means that it is not allocated currently.</p>	1
+DataNodeChannel	int	O	<p>Indicates the local channel of data node to which the global channel is currently allocated (N+0 cluster).</p> <p>Counted from 0. If it is -1 or when the field does not exist, it means that it is not currently allocated.</p>	2
+AutoReset	bool	O	<p>Auto reset.</p> <p>Compatible process:</p> <p>If this parameter is not configured, the default value is false and the channel cannot be reset automatically.</p> <p>when there is no client to access the video stream (streaming client access count changes from non-zero to zero), if AutoReset=true, the device will actively enable the channel to be idle (that is, the video source attribute Enable=false).</p>	false
+VideoStandard	enumchar[32]	O	<p>Current video standard</p> <p>enumchar[32]{</p> <p>"PAL"</p> <p>"NTSC"</p> <p>"SECAM"</p> <p>}</p>	"PAL"
+Label	enumchar[32]	O	<p>Device label</p> <p>enumchar[32]{</p> <p>"Indoor": Indoor</p> <p>"Outdoor": Outdoor</p> <p>}</p>	"Indoor"
+ImageInfo	object	O	Device snapshot	

			When adding a device, take a snapshot and save it to local computer.	
++FilePath	char[256]	R	Image path	"/var/local/ch1.jpg"
+PasswordGenType	enumchar[32]	O	Password generation method enumchar[32]{ "UserInput": Entered by users. "Auto": Automatically generated. }	"UserInput"

Response Example

```
{
  "camera":[{
    "ControlID": "5001",
    "Name": "XXX Road",
    "Enable": true,
    "DeviceInfo":{
      "Enable": true,
      "EncryptStream": 0,
      "Address": "10.6.5.10",
      "Port": 37777,
      "usePreSecret": 1,
      "UserName": "admin",
      "Password": "admin",
      "ProtocolType": "Private",
      "VideoInputChannels": 16,
      "AudioInputChannels": 4,
      "DeviceClass": "IPC",
      "DeviceType": "CP-IPC",
      "HttpPort": 80,
      "HttpsPort": 443,
      "RtspPort": 554,
      "Name": "IPC1",
      "MachineAddress": "XXX District, YYY Road"
      "SerialNo": "Device12345678",
      "VendorAbbr": "VVV",
      "HardID": "454",
      "SoftwareVersion": "2.420.0000006.0.R.150311",
      "ActivationTime": "2000-01-01 00:00:00",
      "NodeType": "",
      "Mac": "08:00:20:0A:8C:6D",
      "OEMVendor": "XXX"
    }
    "DeviceID": "dev123",
    "Type": "Remote",
    "VideoStream": "Main",
    "Channel": 0,
    "UniqueChannel": 0
  }]
}
```

```

    "DataNodeId": 1,
    "DataNodeChannel": 2,
    "AutoReset": false,
    "VideoStandard": "PAL",
    "Label": "Indoor",
    "ImageInfo": {
        "FilePath": "/var/local/ch1.jpg"
    },
    "PasswordGenType": "UserInput"
},...{}]
}

```

4.6.26 Subscribing for Device Online/Offline Status

Subscribe to the online and offline status of the device.

Request URL	http://<server>/cgi-bin/api/LogicDeviceManager/attachCameraState			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
channel	int[]	R	Observed unique channel number. "-1" means all unique channels. If one of the array elements is "-1", it will observe all channels.	[1, 2]
heartbeat	int32	O	Send heartbeat interval, range is [1, 60], unit is second. If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client, the heartbeat message is a string "Heartbeat". If this parameter is not present, its default value is 60.	10
Request Example				
{				
"type": [1, 2],				
"heartbeat": 10				
}				

Response Params (multipart in body)				
Name	Type	R/O	Description	Example
state	object	R	The new status of the camera	
+channel	int32	R	Unique channel number (the channel number with changed status)	1
+connectionState	enumchar[12]	R	Connection status enumchar[12]{	"Connected"

		<p>"Connecting" "Connected" "Unconnect": Not connected "Empty": The channel is not configured and there is no information. "Disable": The channel is configured but disabled (the protocol cannot be reflected, and delete it first). "Changed": The information of the device connected to the channel is changed. "Hibernation" }</p>	
--	--	--	--

Response Example

HTTP/1.1 200 OK

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: close

--<boundary>

Content-Type: application/json

Content-Length: <data length>

{

 "channel" : 1,
 "connectionState" : "Connected"

}

--<boundary>

Content-Type: text/plain

Content-Length: 11

Heartbeat

--<boundary>

...

4.6.27 Get Complete Machine Version

get complete machine version

Request URL	http://<server>/cgi-bin/api/MagicBox/getCompleteMachineVersion
--------------------	--

Method	POST
---------------	------

Request Params (JSON format in body)			
---	--	--	--

Name	Type	R/O	Description	Example
------	------	-----	-------------	---------

Request Example

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
CompleteMachineVersion	char[32]	R	complete machine version	"S2"
Response Example				
{ "CompleteMachineVersion": "S2" }				

4.6.28 Connection test

network connection test of devices and http servers

Request URL	http://<server>/cgi-bin/api/tcpConnect/tcpTest			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Ip	char[40]	R	Http server IP	"10.34.9.21"
Port	int32	R	The port number of the http server	80
Request Example				
{ "Ip": "10.34.9.21", "Port": 80 }				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				
{ "Connect": true }				

4.6.29 Getting Online Status of the Channel

Request URL	http://<server>/cgi-bin/api/LogicDeviceManager/getCameraState			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
uniqueChannel	Int[]	R	Unique channel number If the array is 1, the corresponding value is -1, and it means to get all	[-1]

		the channels.	
Request Example			
{ "uniqueChannels": [-1] }			

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
states	object[]	R	Online status	
+ channel	uint	R	Channel number	0
++connection State	enumchar[12]	R	Connect status enumchar[12]{ "Connecting": connecting. "Connected": successfully connected "Unconnect": configured but not connect "Empty": The channel is not configured or configured into "" "Disable": The channel is configured but disabled. "UnInitiated": the front-end devices are not initialized. "Hibernation": hibernate }	"Connected"
++capsState	bool	O	Represents whether the current channel can get capacity from the front-end devices.	true
++errorMessage	char[128]	O	Reason for failed to connect When the connection status is unconnected, it is valid. "LoginConnectFailed", failed to connect to the internet when login. "ErrorHasNotInit", failed to login for the device is not initialized and there are no users. "UserOrPwdNotValid", failed to login for entering the wrong user name or the wrong password. "LoginConnectTimeout", login connection timeout. "LoginFailure", failed to login for unknown reason.	"LoginConnectFailed"

Response Example			
{ "states" : [{ "channel" : 0, "connectionState" : "Connecting", }			

```

        "capsState": true,
        "errorMessage": "LoginConnectFailed"
    }, ..., {}
}

```

4.6.30 Getting the Recording Status of the Channel

Request URL	http://<server>/cgi-bin/api/recordManager/getStateAll			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Request Example				{}

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
state	object[]	R	Recording status. The array index represents the corresponding channel.	
+ Main	object	R	Main stream	
++ State	uint	R	Whether the main stream is recording, and if yes, Flag has meaning.	1
+Extra1	object	O	If the recording can be performed in sub streams, then it may exist information such as the recording status of the sub streams, and it is the same with field “Main”.	
+Extra2	object	O	If the recording can be performed in sub streams, then it may exist information such as the recording status of the sub streams, and it is the same with field “Main”.	
+Extra3	object	O	If the recording can be performed in sub streams, then it may exist information such as the recording status of the sub streams, and it is the same with field “Main”.	

Response Example
{ "state" : [{ "Main" : { "State" : 1 }, "Extra1" : {}, "Extra2" : {}, "Extra3" : {} }] }

```

        "Extra3" : {}
    }, ..., {}]
}

```

4.7 User management

4.7.1 Get Information of a Particular User

Table 4-42

Syntax	http://<server>/cgi-bin/userManager.cgi?action=getUserInfo&name=< userName >
Method	GET
Description	Get user information with name userName .
Example	http://192.168.1.250/cgi-bin/userManager.cgi?action=getUserInfo&name=admin
Success Return	user.Name=admin user.Memo=admin 's account user.Group=admin user.Reserved=true user.Sharable=true user.PwdValidPeriod=30 user. AuthList=< authList >
Comment	—

4.7.2 Get Information of All Users

Table 4-43

Syntax	http://<server>/cgi-bin/userManager.cgi?action=getUserInfoAll
Method	GET
Description	Get information of all users.
Example	http://192.168.1.250/cgi-bin/userManager.cgi?action=getUserInfoAll
Success Return	users[0].Group=admin users[0].Id=1 users[0].Memo=admin 's account users[0].Name=admin users[0].Reserved=true users[0].Sharable=true users[0].PwdValidPeriod=30 users[0]. AuthList=< authList > users[1].Group=admin ...
Comment	—

4.7.3 Get Information of All Active Users

Table 4-44

Syntax	http://<server>/cgi-bin/userManager.cgi?action=getActiveUserInfoAll
Method	GET
Description	Get active users.
Example	http://192.168.1.250/cgi-bin/userManager.cgi?action=getActiveUserInfoAll
Success Return	users[0].name=admin users[0].ip=10.43.2.16 users[0].group=admin users[0].clienttype=web3.0 users[0].logintime=2011-11-08 09:51:03
Comment	—

4.7.4 Get Information of a Particular Group

Table 4-45

Syntax	http://<server>/cgi-bin/userManager.cgi?action=getGroupInfo&name=< groupName >
Method	GET
Description	Get group setting with name groupName .
Example	http://192.168.1.250/cgi-bin/userManager.cgi?action=getGroupInfo&name=admin
Success Return	group.Name=admin group.Memo=administrator group group.AuthorityList=<authList>
Comment	Parameters in URL: The device has one or two default user groups: "admin" or "admin" and "user". The "admin" group has all the authorities of operating the device. The "user" group only has monitoring and replaying authorities. groupName : name of the group. If the group named groupName does not exist, the device returns Error.

4.7.5 Get Information of All Groups

Table 4-46

Syntax	http://<server>/cgi-bin/userManager.cgi?action=getGroupInfoAll
Method	GET
Description	Get information of all groups.
Example	http://192.168.1.250/cgi-bin/userManager.cgi?action=getGroupInfoAll
Success Return	group[0].Id=1 group[0].Name=admin group[0].Memo=administrator group group[0].AuthorityList=<authList> group[0].AuthorityDetail.Bypass[0]=1

	<pre> group[0].AuthorityDetail.Bypass[1]=2 group[0].AuthorityDetail.Bypass[2]=3 ... group[0].AuthorityDetail.AccessControl[0]=1 group[0].AuthorityDetail.AccessControl[1]=3 group[0].AuthorityDetail.AccessControl[2]=5 ... group[1].Id=1 group[1].Name=user group[1].Memo=user group group[1].AuthorityList=<authList> group[1].... ... </pre>
Comment	—

4.7.6 Add a New User

Access Control products are not supported.

Table 4-47

Syntax	<code>http://<server>/cgi-bin/userManager.cgi?action=addUser&user.Name=<userName>&user.Password=<userPassword>&user.Group=<userGroup>&user.Sharable=<userSharable>[&user.Memo=<userMemo>&user.Reserved=<userReserved>]</code>
Method	GET
Description	Add a user.
Example	<code>http://192.168.1.250/cgi-bin/userManager.cgi?action=addUser&user.Name=Vedant&user.Password=123456&user.Group=user&user.Sharable=true&user.Reserved=false</code>
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>userGroup: string, the range is "admin"and "user". In different group, the user has different authorities.</p> <p>userSharable: bool, true means allow multi-point login.</p> <p>userReserved: bool, true means this user can't be deleted.</p>

4.7.7 Delete a User

Access Control products are not supported.

Table 4-48

Syntax	<code>http://<server>/cgi-bin/userManager.cgi?action=deleteUser&name=<userName></code>
Method	GET
Description	Delete user with name username .
Example	<code>http://192.168.1.250/cgi-bin/userManager.cgi?action=deleteUser&name=Vedant</code>
Success Return	OK

4.7.8 Modify User Information

Access Control products are not supported.

Table 4-49

Syntax	<code>http://<server>/cgi-bin/userManager.cgi?action=modifyUser&name=<UserName>&user.Memo=<userMemo>&user.Group=<userGroup>&user.Reserved=<userReserved>&user.Sharable=<userSharable></code>
Method	GET
Description	Modify user info.
Example	<code>http://192.168.1.250/cgi-bin/userManager.cgi?action=modifyUser&name=Vedant&user.Group=admin</code>
Success Return	OK
Comment	User is identified by <UserName>, other params are the same with AddUser.

4.7.9 Modify User's Password

Access Control products are not supported.

Table 4-50

Syntax	<code>http://<server>/cgi-bin/userManager.cgi?action=modifyPassword&name=<username>&pwd=<newPwd>&pwdOld=<oldPwd></code>
Method	GET
Description	Modify user password.
Example	<code>http://192.168.1.250/cgi-bin/userManager.cgi?action=modifyPassword&name=Vedant&pwd=abcdef&pwdOld=123456</code>
Success Return	OK
Comment	Old password oldPwd should be supplied, new password is newPwd .

4.7.10 Manager Modify Common User's Password

Access Control products are not supported.

Table 4-51

Syntax	<code>http://<server>/cgi-bin/userManager.cgi?action=modifyPasswordByManager&userName=<username>&pwd=<newPwd>&managerName=<managerName>&managerPwd=<managerPwd>&accountType=<accountType></code>
Method	GET
Description	Manager Modify common user's password.
Example	<code>http://192.168.1.250/cgi-bin/userManager.cgi?action=modifyPasswordByManager&userName=Vedant&pwd=123456&managerName=admin&managerPwd=abc123&account</code>

	Type=0
Success Return	OK
Comment	the name of common user to be modified is username new password is newPwd manager name is only admin manager password is managerPwd, that is password of admin accountType support 0 is common account, 1 is Onvif account

4.7.11 [Config] Set User Login Authentication Policy

Table 4-52

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set User Login Authenticate Policy
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&LoginAuthCtrl.PriSvrPolicy=2
Success Return	OK
Comment	LoginAuthCtrl.PriSvrPolicy : login policy setting for Private Protocol. 1 - safe mode, only support digest authentication. (Recommended.); 2 - compatible mode for traditional devices;

4.7.12 Export all the user info

Export all the user info in Excel format

Request URL	http://<server>/cgi-bin/api/userManager/accountFileExport			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
FileType	uint32	是	file type ; 1 : account file 2 : template file	1
Request Example				
{ "FileType": 1 }				

Response Params (binary in body)	
Name	Type
Response Example	
HTTP/1.1 200 OK Server: Device/1.0 Content-Type: application/octet-stream	

Content-Length: <length>

< binary data>

4.8 Network

4.8.1 Get Network Interfaces

Table 4-53

Syntax	http://<server>/cgi-bin/netApp.cgi?action=getInterfaces
Method	GET
Description	Get all of the system network interfaces.
Example	http://192.168.1.250/cgi-bin/netApp.cgi?action=getInterfaces
Success Return	netInterface[0]. Name =eth0 netInterface[0]. Type =Normal netInterface[0]. Valid =true ... netInterface[1]. Name =3g netInterface[1]. Type =Auto,WCDMA,TD-SCDMA,TD-LTE,FDD-LTE netInterface[1]. Valid =true ...
Comment	<p>result item value:</p> <p>Name: network interface name. "eth0" - wired network interface "eth2" - wireless network interface "3G" - 3G network interface "bond0" - bond network interface "bond1" - bond network interface Not listed in detail</p> <p>Type: Network types that support settings "Normal" — wired network "Wireless" — wireless network "Auto", "TD-SCDMA", "WCDMA", "CDMA1x", "EDGE", "EVDO", "TD-LTE", "FDD-LTE", "NR" — 3G network types.</p> <p>Valid: network interface is valid if netInterface[n].Valid is true.</p>

4.8.2 Get Client Access Filter

Table 4-54

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=AccessFilter
Method	GET

Description	Get access filter config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=AccessFilter
Success Return	table.AccessFilter.Enable=false table.AccessFilter.TrustList[0]=10.6.10.23 table.AccessFilter.TrustList[1]=10.6.10.62 table.AccessFilter.Type=TrustList
Comment	Parameters in Response: Type: Range is {TrustList, BannedList}. If Trustlist is supported, Bannedlist is not supported. More information in Appendix.

Appendix:

ParamName	ParamValue type	Description
AccessFilter.BannedList[<i>index</i>]	string	Banned IP address list
AccessFilter.TrustList[<i>index</i>]	string	Trusted IP address list
AccessFilter.Enable	bool	Enable/Disable access filter function
AccessFilter.Type	string	Range is {TrustList, BannedList}, TrustList: Trust list is used, banned list is not used. BannedList: Banned list is used, trust list is not used.

4.8.3 [Config] Network Basic Config

- Get network config

Table 4-55

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=Network
Method	GET
Description	Get network basic config. The basic config contains basic network parameters (default interface, domain name, host name), and configuration of each network interface.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=Network
Success Return	table.Network.DefaultInterface=eth0 table.Network.Domain=ttt table.Network.Hostnamehhh table.Network. <i>interface</i> .DefaultGateway=10.7.0.1 table.Network. <i>interface</i> .DhcpEnable=false table.Network. <i>interface</i> .DnsServers[0]=221.123.33.228 table.Network. <i>interface</i> .DnsServers[1]=221.12.1.228 table.Network. <i>interface</i> .IPAddress=10.7.2.3 table.Network. <i>interface</i> .MTU=1500 table.Network. <i>interface</i> .PhysicalAddress=00:10:5c:f2:1c:b4 table.Network. <i>interface</i> .SubnetMask=255.255.0.0
Comment	interface in response is network interface name, such as eth0, eth2...

- Set network config

Table 4-56

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set network basic config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&NetWork.Domain=ttt&NetWork.eth0.DhcpEnable=true
Success Return	OK
Comment	interface in below ParamName is network interface name, such as eth0 and eth2

Appendix:

ParamName	ParamValue type	Description
Network.DefaultInterface	string	Set default network interface when multiple interfaces exist. Range of interfaces is depends on GetInterfaces .
Network.Domain	string	Domain name.
Network.Hostname	string	Hostname and Domain compose a network address.
Network. interface .DefaultGateway	string	IP address.
Network. interface .DhcpEnable	bool	Enable/Disable DHCP.
Network. interface .DnsServers[0]	string	IP address of first DNS server.
Network. interface .DnsServers[1]	string	IP address of second DNS server.
Network. interface .IPAddress	string	Interface IP address.
Network. interface .MTU	integer	Interface MTU.
Network. interface .PhysicalAddress	string	MAC address of interface. HEX string in the form of: xx:xx:xx:xx:xx:xx. Range of x is [0-9, a-f, A-F] Example: 00:10:5c:f2:1c:b4 00:10:5C:F2:1C:B5
Network. interface .SubnetMask	string	Network mask string: In the form of x.x.x.x, range of x is [0-255] Example: 255.255.255.0
Network. interface .EnableDhcpReservedIP	bool	Enable/Disable when dhcp failed, it will continue sending dhcp request, if it was enabled.

4.8.4 [Config] PPPoE

- Get PPPoE config

Table 4-57

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=PPPoE
Method	GET
Description	Get PPPoE config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=PPPoE
Success Return	table.PPPoE.Enable=true, table.PPPoE.AuthType[0]=PAP, table.PPPoE.UserName=hz150260, table.PPPoE.Password=Dont Know, table.PPPoE.Mode=Default, table.PPPoE.Eth=eth0, table.PPPoE.DefaultPPPoEInterface=ppp0, table.PPPoE.PPPOE1.Enable=true, table.PPPoE.PPPOE1.AuthType[0]=PAP, table.PPPoE.PPPOE1.UserName=hz150260, table.PPPoE.PPPOE1.Password=Don't Know, table.PPPoE.PPPOE1.Mode=Custom, table.PPPoE.PPPOE1.Eth=eth1
Comment	—

- Set PPPoE config

Table 4-58

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue> [&<paramName>=<paramValue> ...]
Method	GET
Description	Set PPPoE config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&PPPoE.UserName=user1&PPPoE.Password=123456
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
PPPoE.Enable	bool	Enable/Disable PPPoE.
PPPoE.UserName	string	PPPoE user name.
PPPoE.Password	string	PPPoE user password.

4.8.5 [Config] DDNS

- Get DDNS config

Table 4-59

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=DDNS
--------	--

Method	GET
Description	Get DDNS config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=DDNS
Success Return	table.DDNS[<i>index</i>].Address=www.ttt.com table.DDNS[<i>index</i>].Enable=true table.DDNS[<i>index</i>].HostName=www.ttt.com table.DDNS[<i>index</i>].KeepAlive=10 table.DDNS[<i>index</i>].Password=none table.DDNS[<i>index</i>].Port=5050 table.DDNS[<i>index</i>].Protocol= Quick DDNS table.DDNS[<i>index</i>].UserName=user1 table.DDNS[<i>index</i>].DefaultHostName.Enable=false table.DDNS[<i>index</i>].DefaultHostName.HostName=9002A9D77133.ttt.com
Comment	<i>index</i> in response is the DDNS protocol table index, start from 0. The meaning of parameters can refer to SetDDNSConfig chapter.

- Set DDNS config

Table 4-60

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set DDNS config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&DDNS[0].Address=www.ttt.com&DDNS[0].Enable=true
Success Return	OK
Comment	<i>index</i> in below ParamName is the DDNS protocol table index, start from 0.

Appendix:

ParamName	ParamValue type	Description
DDNS[<i>index</i>].Address	string	DDNS server IP address or name.
DDNS[<i>index</i>].Enable	bool	Multiple DDNS hostname can be configured, but Only one hostname can be enabled, others should not be enabled.
DDNS[<i>index</i>].HostName	String	Hostname of this device.
DDNS[<i>index</i>].KeepAlive	integer	Range is [1—65535]. Unit is minutes.
DDNS[<i>index</i>].Password	string	DDNS user password
DDNS[<i>index</i>].Port	integer	Range is [1—65535]. Port of DDNS server

ParamName	ParamValue type	Description
DDNS[<i>index</i>].Protocol	string	<p>DDNS protocol type. Range is { "PRIVATE DDNS": private ddns version 1 "Private DDNS": private ddns version 2 "IND99 DDNS" "NO-IP DDNS" "Dyndns DDNS" "LUPUS DDNS" "Oray DDNS" "CP Plus DDNS" "Ipplus DDNS" "Q-See DDNS" "SYSDNS DDNS" "Videotrend DDNS" "CEPSA DDNS" "G4IP DDNS" "HOSS DDNS" "Intelbras DDNS" "HSY DDNS" "Flir DDNS" "Private P2P DDNS": private ddns for P2P "MiVigilante DDNS" "ByDemes DDNS" "DYNDNS OLD DDNS": old dyndns for storage device "WATASHI DDNS" "INDB DDNS" "Peoplefu DDNS" "MINT DDNS" "SPECO DDNS" "SHANY DDNS" "WITHCCTV DDNS": "SONIC DDNS" "KBVISION DDNS" "BOSCH DDNS" }.</p>
DDNS[<i>index</i>].UserName	string	DDNS user name
DDNS[<i>index</i>].DefaultHostName .Enable	bool	Only protocol is in range {"Private DDNS", "DHDDNS", "QUICK DDNS"}, it effects. true : use the DefaultHostName.HostName false: use the HostName
DDNS[<i>index</i>].DefaultHostName .HostName	string	The default hostname. It cannot be modified.

4.8.6 [Config] Email

- Get email config

Table 4-61

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=Email
Method	GET
Description	Get Email config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=Email
Success Return	table.Email.Address=www.ttt.com table.Email.Anonymous=true table.Email.AttachEnable=true table.Email.Enable=true table.Email.HealthReport.Enable=false table.Email.HealthReport.Interval=61 table.Email.Password=123456 table.Email.Port=26 table.Email.Receivers[0]=x@ttt.com table.Email.Receivers[1]=y@ttt.com table.Email.Receivers[2]=z@ttt.com table.Email.SendAddress=x@ttt.com table.Email.SslEnable=false table.Email.TlsEnable=false table.Email.Authentication=false table.Email.OnlyAttachment=false table.Email.SendInterv=10 table.Email.Title=DVRMessage table.Email.UserName=anonymity
Comment	—

- Set email config

Table 4-62

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue> [&<paramName>=<paramValue> ...]
Method	GET
Description	Set Email config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&Email.Address=mail.ttt.com&Email.Anonymous=false
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
Email.Address	string	SMTP server IP address or name.

ParamName	ParamValue type	Description
Email.Anonymous	bool	Enable/Disable anonymous email.
Email.AttachEnable	bool	Enable/Disable email attachment
Email.AttachmentEnable	bool	Enable/Disable email attachment
Email.Enable	bool	Enable/Disable email function
Email.HealthReport.Enable	bool	Enable/Disable report device status by email.
Email.HealthReport.Interval	integer	Range is [30-1440]. Unit is minutes
Email.Password	string	User password of email account.
Email.Port	integer	Range is [1-65535]
Email.Receivers[0]	string	Email addresses of 3 receivers.
Email.Receivers[1]	string	
Email.Receivers[2]	string	
Email.SendAddress	string	Sender email address.
Email.SslEnable	bool	True: enable SSL email.
Email.Title	string	Title of email.
Email.UserName	string	User name of email account.

4.8.7 [Config] WLan

- Get WLan config

Table 4-63

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=WLan
Method	GET
Description	Get Wlan config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=WLan
Success Return	table.WLan.eth2.Enable=true table.WLan.eth2.Encryption=off table.WLan.eth2.KeyFlag=false table.WLan.eth2.KeyID=0 table.WLan.eth2.KeyType=Hex table.WLan.eth2.Keys[0]=password1 table.WLan.eth2.Keys[1]=password2 table.WLan.eth2.Keys[2]=password3 table.WLan.eth2.Keys[3]=password4 table.WLan.eth2.LinkMode=Auto table.WLan.eth2.SSID=ttt
Comment	—

- Set WLan config

Table 4-64

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET

Description	Set WLan config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&WLan.eth2.Enable=true&WLan.eth2.KeyType=Hex
Success Return	OK
Comment	In below ParamName, interface is name of wireless interface.

Appendix:

ParamName	ParamValue type	Description
WLan. interface .Enable	bool	True: Enable WLan on this interface.
WLan. interface .Encryption	string	Range is {Off, On, WEP64Bits, WEP128Bits, WPA-PSK-TKIP, WPA-PSK-CCMP} Encryption mode.
WLan. interface .KeyFlag	bool	true: key is configured.
WLan. interface .KeyID	integer	Range is [0—3] Indicates which key is used. 0: WLan. interface .Keys[0] is used.
WLan. interface .KeyType	string	Range is {Hex, ASCII}
WLan. interface .Keys[0]	string	For ASCII key type: 64bits encryption key length is 5, 128bits encryption key length is 13, consists of [0—9, a—z, A—Z]
WLan. interface .Keys[1]	string	For HEX key type: 64bits encryption key length is 10, 128bits encryption key length is 26, consists of [0—9, a—z, A—Z]
WLan. interface .Keys[2]	string	
WLan. interface .Keys[3]	string	
WLan. interface .LinkMode	string	Range is {Auto, Ad-hoc, and Infrastructure}. Auto – select suitable mode automatically. Ad-hoc – Device with wireless network adapter can connect to each other without Access Point. Infrastructure – Integrate wire and wireless LAN together to share network resource, access point is need in this mode.
WLan. interface .SSID	string	SSID.

4.8.8 Scan Wlan Devices

Table 4-65

Syntax	http://<server>/cgi-bin/wlan.cgi?action=scanWlanDevices&<paramName>=<paramValue> [&<paramName>=<paramValue> ...]
Method	GET
Description	Search Wi-Fi device information.
Example	http://192.168.1.250/cgi-bin/wlan.cgi?action=scanWlanDevices&SSID=xia_yuguo 13098 Internet
Success Return	found=1 wlanDevice[0].ApConnected=0 wlanDevice[0].ApMaxBitRate=54000000

	wlanDevice[0].ApNetWorkType=255 wlanDevice[0].AuthMode=7 wlanDevice[0].BSSID=28:2c:b2:5c:de:36 wlanDevice[0].EncrAlgr=3 wlanDevice[0].LinkMode=0 wlanDevice[0].LinkQuality=31 wlanDevice[0].RSSIQuality=0 wlanDevice[0].SSID=xia_yuguo 13098 Internet
Comment	—

Appendix:

ParamName	ParamValue type	Description
SSID	string	Specified SSID, if not include any SSID, all Wi-Fi information will be searched and displayed.

4.8.9 [Config] UPnP

- Get UPnP config

Table 4-66

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=UPnP
Method	GET
Description	Get UPnP config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=UPnP
Success Return	table.UPnP.Enable=true table.UPnP.MapTable[<i>index</i>].Enable=true table.UPnP.MapTable[<i>index</i>].InnerPort=80 table.UPnP.MapTable[<i>index</i>].OuterPort=8080 table.UPnP.MapTable[<i>index</i>].Protocol=TCP table.UPnP.MapTable[<i>index</i>].ServiceName=HTTP
Comment	<i>index</i> in response is the UPNP map table index, start from 0.

- Set UPnP config

Table 4-67

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue> [&<paramName>=<paramValue> ...]
Method	GET
Description	Set UPnP config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&UPnP.Enable=true&UPnP.MapTable[0].Protocol=TCP
Success Return	OK
Comment	index in below ParamName is UPNP map table index, range is [0-255]

Appendix:

ParamName	ParamValue type	Description
UPnP.Enable	bool	Enable/Disable UPNP feature.
UPnP.MapTable[<i>index</i>].Enable	bool	Enable/Disable this UPNP map.
UPnP.MapTable[<i>index</i>].InnerPort	integer	Range is [1—65535]. Inner port number
UPnP.MapTable[<i>index</i>].OuterPort	integer	Range is [1—65535]. Outer port number.
UPnP.MapTable[<i>index</i>].Protocol	string	Range is {TCP, UDP}
UPnP.MapTable[<i>index</i>].ServiceName	string	User defined UPnP service name.

4.8.10 Get UPnP Status

Table 4-68

Syntax	http://<server>/cgi-bin/netApp.cgi?action=getUPnPStatus
Method	GET
Description	Get UPnP Status.
Example	http://192.168.1.250/cgi-bin/netApp.cgi?action=getUPnPStatus
Success Return	status.InnerAddress=0.0.0.0 status.OuterAddress=0.0.0.0 status.PortMapStatus[0]=Failed status.PortMapStatus[1]=Failed status.PortMapStatus[2]=Failed status.PortMapStatus[3]=Failed status.Status=Unknown status.Working=false
Comment	—

4.8.11 [Config] NTP

- Get NTP config

Table 4-69

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=NTP
Method	GET
Description	Get NTP config
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=NTP
Success Return	table.NTP.Address=clock.isc.org table.NTP.Enable=false table.NTP.Port=38 table.NTP.TimeZone=9 table.NTP.UpdatePeriod=31 table.NTP.ServerList[0].Enable=true, table.NTP.ServerList[0].Address=192.168.1.250, table.NTP.ServerList[0].Port=123 table.NTP.Tolerance=5,

	table.NTP.TimeZoneDesc=GMT
Comment	—

- Set NTP config

Table 4-70

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set NTP config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&NTP.Address=time.ttt.com&NTP.Enable=true
Success Return	OK
Comment	—

4.8.12 [Config] RTSP

- Get RTSP config

Table 4-71

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=RTSP
Method	GET
Description	Get RTSP config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=RTSP
Success Return	table.RTSP.Enable=true table.RTSP.Port=554 table.RTSP.RTP.EndPort=40000 table.RTSP.RTP.StartPort=20000
Comment	—

- Set RTSP config

Table 4-72

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set RTSP config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&RTSP.Enable=true&RTSP.Port=554
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
-----------	-----------------	-------------

ParamName	ParamValue type	Description
RTSP.Enable	bool	Enable/Disable RTSP.
RTSP.Port	integer	RTSP port.
RTSP.RTP.StartPort	integer	RTP start port.
RTSP.RTP.EndPort	integer	RTP end port.

4.8.13 [Config] Alarm Server

- Get alarm server config

Table 4-73

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=AlarmServer
Method	GET
Description	Get alarm server config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=AlarmServer
Success Return	table.AlarmServer.Address=10.7.8.9 table.AlarmServer.Enable=false table.AlarmServer.Password= table.AlarmServer.Port=8888 table.AlarmServer.Protocol=ttt table.AlarmServer.ReportTime=02:00:00 table.AlarmServer.ReportWeekDay=2 table.AlarmServer.UserName=admin
Comment	—

- Set alarm server config

Table 4-74

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set Alarm Server config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&AlarmServer.Address=as.ttt.com&AlarmServer.Enable=false
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
AlarmServer.Address	string	Alarm server IP address or name.
AlarmServer.Enable	bool	Enable/Disable Alarm server.
AlarmServer.Port	integer	Range is [1—65535]. Port of Alarm server.

4.8.14 [Config] Onvif Service Authorization

- Get config of Onvif service authorization

Table 4-75

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=UserGlobal
Method	GET
Description	Get user global config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=UserGlobal
Success Return	table.UserGlobal.OnvifLoginCheck=false
Comment	If "OnvifLoginCheck" is false, you can get Onvif service directly; if true, you should enter your ID/username and password.

- Set config of Onvif service authorization

Table 4-76

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&UserGlobal.OnvifLoginCheck=<flag>
Method	GET
Description	Enable onvif login check or not.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&UserGlobal.OnvifLoginCheck=true
Success Return	OK
Comment	Parameters in URL: flag : range is {true, false}.

4.8.15 [Config] SSHD Config

- Get SSHD config

Table 4-77

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&&name=SSHD
Method	GET
Description	Get SSHD Config
Example	http://<server>/192.168.1.250/configManager.cgi?action=getConfig&&name=SSHD
Success Return	table.SSHD.Enable=true table.SSHD.Port=22
Comment	—

- Set SSHD config

Table 4-78

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramVal>
--------	---

	ue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set SSHD Config
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&SSHD.Enable=true&SSHD.Port=22
Success Return	OK
Comment	—

4.8.16 [Config] Cellular Network Traffic Packages

Parameters:

Config Data Params

Name	Type	R/O	Description	Example
CellularFlux	object	O	Configure cellular network traffic packages.	
+3G	object	O	Connect cellular network. Supports 4 cellular modules. Single module: 3G; multiple modules: 3G, 3G1, 3G2, and 3G3. Notes: Considering the compatibility, the 3G naming is reserved, which can actually represent 4G and 5G network adapters.	
++SIM1	object	O	Configure the traffic package of the first SIM card. If there is a second one, it will be named SIM2, and the third one will be named SIM3.	
+++Enable	bool	O	Enable traffic strategy.	false
+++MonthFlux	object	O	Configure monthly traffic.	
++++FluxUp	uint32	O	The upper limit of traffic usage	299
++++FluxUpUnit	char[32]	O	The upper limit unit of traffic usage. GB and MB are available.	"GB"
++++FluxWarn	uint32	O	The upper limit of traffic reminders	99
++++FluxWarnUnit	char[32]	O	The upper limit unit of traffic reminders. GB and MB are available.	"GB"
++++FluxStartDay	uint32	O	The digit indicates what day the package starts every month.	1

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=CellularFlux
```

Get Config Response Example

```
table.CellularFlux.3G.SIM1.Enable=true
table.CellularFlux.3G.SIM1.MonthFlux.FluxStartDay=1
table.CellularFlux.3G.SIM1.MonthFlux.FluxUp=299
table.CellularFlux.3G.SIM1.MonthFlux.FluxUpUnit=GB
```

table.CellularFlux.3G.SIM1.MonthFlux.FluxWarn=99
table.CellularFlux.3G.SIM1.MonthFlux.FluxWarnUnit=GB

Set Config Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&CellularFlux.3G.SIM1.Enable=true&CellularFlux.3G.SIM1.MonthFlux.FluxStartDay=1&CellularFlux.3G.SIM1.MonthFlux.FluxUp=299&CellularFlux.3G.SIM1.MonthFlux.FluxUpUnit=GB&CellularFlux.3G.SIM1.MonthFlux.FluxWarn=99&CellularFlux.3G.SIM1.MonthFlux.FluxWarnUnit=GB
```

Set Config Response Example

OK

4.8.17 Obtaining the Traffic Information of Current Month

Acquiring the traffic information of current month

Request URL	http://<server>/cgi-bin/api/DataFlux/getFlux			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
card	char[32]	R	The key value of the first-level node under Wireless. For earlier devices, it represents the SIM card number; For new devices, it represents the module name. The specific values are: enumchar[32]{ 3G 3G1 3G2 3G3 } Single card/module: 3G; Multiple cards/modules: 3G, 3G1, 3G2, and 3G3.	"3G"
SimName	char[32]	O	SIM card name. The first SIM card is SIM1, the second SIM2, and the third SIM3.	"SIM1"
ICCID	char[24]	O	IC card identification code (namely SIM card number). It consists of 20 digits.	
Request Example				
{				

```

"card": "3G",
"SimName": "SIM1",
"ICCID": "898607E2112251029357"
}

```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
info	object	R	Traffic information.	
+LastMonthTotalFlux	uint	R	The total traffic of the last month. Unit: 0.1 kb.	8000
+totalFlux	uint	R	Total traffic = uplink traffic + downlink traffic. Unit: 0.1 kb.	8000
+SendFlux	uint	R	Uplink traffic. Unit: 0.1 kb.	3000
+RecvFlux	uint	R	Downlink traffic. Unit: 0.1 kb.	5000
+Time	string YYYY-MM-DD hh:mm:ss	R	Total traffic write time.	"2014-01-01 14:01:01" "
+Record	object[]	R	Traffic records in the last seven days.	
++Date	string	R	Date.	"2014-01-01"
++DaySendFlux	uint	R	Uplink traffic of the day. Unit: 0.1 kb.	3000
++DayRecvFlux	uint	R	Downlink traffic of the day. Unit: 0.1 kb.	3000
++DayTotalFlux	uint64	O	The total traffic of the day. Unit: Byte.	8000
++MonthTotalFlux	uint64	O	The total traffic of the month. Unit: Byte.	8000

Response Example
<pre>{ "info": { "LastMonthTotalFlux" : 8000, "totalFlux" : 8000, "SendFlux": 3000, "RecvFlux": 5000, "Time": "2014-01-01 14:01:01", "Record" : [{"Date": "2014-01-01", "DaySendFlux" : 3000, "DayRecvFlux" : 3000, "DayTotalFlux" : 8000, "MonthTotalFlux" : 8000 }, ...] } }</pre>

}

4.8.18 Searching for History Traffic Statistics According to Specified Conditions

Searching for history traffic statistics by specified conditions.

Request URL	http://<server>/cgi-bin/api/DataFlux/queryHistoryFlux			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
condition	object	R	Search condition	
+Card	char[32]	R	<p>The key value of the first-level node under Wireless. For earlier devices, it represents the SIM card number; For new devices, it represents the module name. The specific values are:</p> <pre>enumchar[32]{ 3G 3G1 3G2 3G3 }</pre> <p>Single card/module: 3G; Multiple cards/modules: 3G, 3G1, 3G2, and 3G3.</p>	"3G"
+SimName	char[32]	O	SIM card name. The first SIM card is SIM1, the second SIM2, and the third SIM3.	"SIM1"
+ICCID	char[24]	O	IC card identification code (namely SIM card number). It consists of 20 digits.	"89860116836014532534"
Mode	char[32]	R	<p>Search modes:</p> <pre>enumchar[32]{ month: Search by month day: Search by day hour: Search by hour }</pre>	"day"

+StartTime	char[20]	O	Start time of statistics	"2021-01-01 00:00:00"
+EndTime	char[20]	O	End time of statistics	"2021-04-01 00:00:00"

Request Example

```
{
  "condition": {
    "Card": "3G",
    "Mode": "month",
    "StartTime": "2022-08-01 00:00:00",
    "EndTime": "2022-09-30 23:59:59",
    "SimName": "SIM1",
    "ICCID": "898607E2112251029357"
  }
}
```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
info	object	O	Search results	
+TotalFlux	uint64	R	Total traffic = uplink traffic + downlink traffic. Unit: 0.1 Byte.	1299700
+SendFlux	uint64	R	The total uplink traffic in the statistical time interval. Unit: Byte.	70305
+RecvFlux	uint64	R	The total downlink traffic in the statistical time interval. Unit: Byte.	1229395
+Time	string YYYY-MM-DD hh:mm:ss	R	Data update time.	"2021-03-31 00:00:00" "
+Record	object[]	R	Traffic statistic records. maximum 31 entries for variable length array.	
++Date	string YYYY-MM-DD	R	Year/month/day.	"2014-01-01"
++SendFlux	uint64	R	Uplink traffic of the year/month/day. Unit: Byte.	70305
++RecvFlux	uint64	R	Uplink traffic of the year/month/day. Unit: Byte.	1229395
++TotalFlux	uint64	O	The total traffic in the current statistical period. Unit: Byte.	8000
++StartTime	string	O		"2010-05-25 00:00:00"

	YYYY-MM-DD hh:mm:ss		Start time of this entry.	"
Response Example				
{	"info": {			
	"TotalFlux": 1299700,			
	"SendFlux": 70305,			
	"RecvFlux": 1229395,			
	"Time": "2021-03-31 00:00:00",			
	"Record": [
	{			
	"Date": "2021-03-31",			
	"StartTime": "2022-09-01 00:00:00",			
	"SendFlux": 70305,			
	"RecvFlux": 1229395,			
	"TotalFlux": 0			
	},...{}]			
	}			
}				

4.9 Event

4.9.1 [DataType] Event Handler

- Get event handler config

Table 4-79

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=< handlerName >
Method	GET
Description	Get event handler settings.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=Alarm[0].EventHandler
Success Return	<p>handlerName.EventHandler.AlarmOutChannels[0]=1 handlerName.EventHandler.AlarmOutChannels[1]=1 ... handlerName.EventHandler.AlarmOutEnable=false handlerName.EventHandler.AlarmOutLatch=10 handlerName.EventHandler.BeepEnable=true handlerName.EventHandler.Dejitter=0 handlerName.EventHandler.Delay=30 handlerName.EventHandler.LogEnable=true handlerName.EventHandler.MailEnable=true handlerName.EventHandler.PtzLink[0][0]=None handlerName.EventHandler.PtzLink[0][1]=0</p>

	<pre> handlerName.EventHandler.PtzLink[1][0]=None handlerName.EventHandler.PtzLink[1][1]=0 ... handlerName.EventHandler.PtzLinkEnable=false handlerName.EventHandler.RecordChannels[0]=1 handlerName.EventHandler.RecordChannels[1]=2 ... handlerName.EventHandler.RecordEnable=true handlerName.EventHandler.RecordLatch=10 handlerName.EventHandler.SnapshotChannels[0]=1 handlerName.EventHandler.SnapshotChannels[1]=1 ... handlerName.EventHandler.SnapshotEnable=false handlerName.EventHandler.SnapshotPeriod=3 handlerName.EventHandler.SnapshotTimes=0 handlerName.EventHandler.TimeSection[0][0]=1 01:00:00-24:00:00 handlerName.EventHandler.TimeSection[0][1]=1 01:00:00-24:00:00... ... handlerName.EventHandler.TimeSection[6][5]=1 01:00:00-24:00:00 handlerName.EventHandler.TipEnable=true handlerName.EventHandler.ExAlarmOutEnable=true handlerName.EventHandler.Enable=true, handlerName.EventHandler.FilckerLightType=WhiteLight, handlerName.EventHandler.LightLinkType=Filcker, handlerName.EventHandler.FilckerIntevalTime=5, handlerName.EventHandler.FilckerTimes=5, handlerName.EventHandler.LightDuration=10, handlerName.EventHandler.WhiteLightTimeSection=, handlerName.EventHandler.LightBright=50 handlerName.ExAlarmOutChannels[0] =2 handlerName.ExAlarmOutChannels[1]=3 ... </pre>
Comment	<p>Parameters in URL:</p> <p>handlerName can be one of below four formats:</p> <p>Alarm[ChannelNo].EventHandler</p> <p>MotionDetect[ChannelNo].EventHandler</p> <p>BlindDetect[ChannelNo].EventHandler</p> <p>LossDetect[ChannelNo].EventHandler</p>

- Set event handler config

Table 4-80

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&< paramName >=< paramValue >[&< paramName >=< paramValue >...]
Method	GET
Description	Modify event handler settings.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&Alarm[0].EventHandl

	er.AlarmOutChannels[0]=1&Alarm[0].EventHandler.AlarmOutEnable=true
Success Return	OK
Comment	In below paramName, Meaning of handlerName is the same with GetEventHandler.

Appendix:

paramName	paramValue type	Description
handlerName .EventHandler.AlarmOutChann els[ch]	integer	<p>ch is alarm out channel index which starts from 0. Range is {0, 1},</p> <p>0 — do not output alarm at alarm out channel ch</p> <p>1 — output alarm at alarm out channel ch</p>
handlerName .EventHandler.AlarmOutEnable	bool	Enable/Disable alarm out function.
handlerName .EventHandler.AlarmOutLatch	Integer	Range is [10—300]. Unit is seconds, indicates the time to output alarm after input alarm is cleared.
handlerName .EventHandler.BeepEnable	bool	Enable/Disable beep.
handlerName .EventHandler.Dejitter	integer	Range is [0—255]. Alarm signal dejitter seconds. Alarm signal change during this period is ignored.
handlerName .EventHandler.Delay	integer	Range is [0—300]. Delay seconds before setting take effect.
handlerName .EventHandler.LogEnable	bool	Enable/Disable log for alarm.
handlerName .EventHandler.MailEnable	bool	Enable/Disable mail send for alarm.
handlerName .EventHandler.PtzLink[ch][0]	string	<p>Range is {None, Preset, Tour, Pattern}</p> <p>This is PTZ action linked with events.</p> <p>ch is PTZ channel index which starts from 0.</p>
handlerName .EventHandler.PtzLink[ch][1]	integer	<p>This is the parameter of PtzLink[ch][0],</p> <p>If PtzLink[ch][0] is Preset: this is preset point.</p> <p>Tour: this is tour path number.</p> <p>Pattern: this is pattern number.</p>
handlerName .EventHandler.PtzLinkEnable	Bool	Enable/Disable PTZ link.
handlerName .EventHandler.RecordChannels	array<integer>	record channel list
handlerName .EventHandler.RecordEnable	bool	Enable/Disable record function.

paramName	paramValue type	Description
handlerName .EventHandler.RecordLatch	integer	Range is [10—300]. Unit is seconds, indicates the time to record after input alarm is cleared.
handlerName .EventHandler.SnapshotChannels	array<integer>	snapshot channel list
handlerName .EventHandler.SnapshotEnable	bool	Enable/Disable snapshot function.
handlerName .EventHandler.SnapshotPeriod	integer	Range is [0—255]. Frames between snapshots. 0 means continuously snapshot for every frame.
handlerName .EventHandler.SnapshotTimes	integer	Range is [0—65535] Snapshot times before stop, 0 means don't stop snapshot.
handlerName .EventHandler.TimeSection[wd][ts]	String	<p>It's an effective time period for eventHanlder everyday.</p> <p>wd (week day) range is [0—6] (Sunday—Saturday)</p> <p>ts (time section) range is [0-23], it's index of time section table.</p> <p>Format: mask hh:mm:ss-hh:mm:ss</p> <p>Mask: {0,1}, hh: [0—24], mm: [00—59], ss: [00—59]</p> <p>Mask 0: this time section is not used.</p> <p>Mask 1: this time section is used.</p> <p>Example: TimeSection[1][0]=1 12:00:00-18:00:00 Means EventHandler is effective between 12:00:00 and 18:00:00 at Monday.</p>
handlerName .EventHandler.TipEnable	bool	Enable/Disable local message box tip.
handlerName .EventHandler.ExAlarmOutEnable	bool	Enable/Disable extend alarm out ability
handlerName .EventHandler.LightingLink.Enable	bool	Enable/Disable LightingLink
handlerName .EventHandler.LightingLink.FilckerLightType	String	Filcker Light Type range : "WhiteLight"
handlerName .EventHandler.LightingLink.LightLinkType	string	range : ["Filcker", "KeepLighting"]

paramName	paramValue type	Description
handlerName .EventHandler.LightingLink.FilckerIntervalTime	float	Filcker Interval Time unit: 0.1s
handlerName .EventHandler.LightingLink.FilckerTimes	integer	Filcker Times
handlerName .EventHandler.LightingLink.LightDuration	integer	Light Duration unit: s
handlerName .EventHandler.LightingLink.WhiteLightTimeSection	array<array<string>>	
handlerName .EventHandler.LightingLink.LightBright	integer	brightness
handlerName .ExAlarmOutChannels[ch]	integer	extend alarm out channels

4.9.2 [Config] Alarm Event

- Get alarm config

Table 4-81

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=Alarm
Method	GET
Description	Get alarm config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=Alarm
Success Return	table.Alarm[0].Enable=false table.Alarm[0].EventHandler....(output of EventHandler is described in GetEventHandler) table.Alarm[0].Name=Door1 table.Alarm[0].SensorType=NC table.Alarm[1].... ...
Comment	—

- Set alarm config

Table 4-82

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue> [&<paramName>=<paramValue> ...]
Method	GET
Description	Set alarm config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&Alarm[0].Enable=true
Success Return	OK
Comment	In below ParamName, input is external alarm input channel which starts from 0. EventHandler defines parameter of relevant actions when alarm or event happens. It's also used in following sections about events.

Appendix:

ParamName	ParamValue type	Description
Alarm[<i>input</i>].Enable	bool	Enable/Disable alarm from a input channel
Alarm[<i>input</i>].EventHandler	—	Setting of EventHandler is described in <u>SetEventHandler</u> .
Alarm[<i>input</i>].Name	string	Name of alarm input channel.
Alarm[<i>input</i>].SensorType	string	Range is {NC, NO}. NC: normal close NO: normal open

4.9.3 [Config] Alarm Out

- Get alarm out config

Table 4-83

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=AlarmOut
Method	GET
Description	Get alarm out config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=AlarmOut
Success Return	table.AlarmOut[<i>alarmOutChannel</i>].Mode=0 table.AlarmOut[<i>alarmOutChannel</i>].Name=Beep
Comment	Parameters in Response: <i>alarmOutChannel</i> : the alarm out channel index which starts from 0.

- Set alarm out config

Table 4-84

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set alarm out config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&AlarmOut[0].Mode=0 &AlarmOut[0].Name=port1
Success Return	OK
Comment	<i>port</i> in below ParamName is alarm out port index, start form 0.

Appendix:

ParamName	ParamValue type	Description
AlarmOut[<i>port</i>].Mode	integer	Range is {0, 1, 2} 0: automatically alarm 1: force alarm 2: close alarm
AlarmOut[<i>port</i>].Name	string	Alarm out port name.

4.9.4 Get Alarm Input Channels

Table 4-85

Syntax	http://<server>/cgi-bin/alarm.cgi?action=getInSlots
Method	GET
Description	Get alarm input channel number.
Example	http://192.168.1.250/cgi-bin/alarm.cgi?action=getInSlots
Success Return	result=2
Comment	—

4.9.5 Get Alarm Output Channels

Table 4-86

Syntax	http://<server>/cgi-bin/alarm.cgi?action=getOutSlots
Method	GET
Description	Get alarm output channel number.
Example	http://192.168.1.250/cgi-bin/alarm.cgi?action=getOutSlots
Success Return	result=1
Comment	—

4.9.6 Get States of Alarm Input Channels

Table 4-87

Syntax	http://<server>/cgi-bin/alarm.cgi?action=getInState
Method	GET
Description	Get alarm input state for all channels.
Example	http://192.168.1.250/cgi-bin/alarm.cgi?action=getInState
Success Return	result=3
Comment	A bit in the response result indicates a channel alarm states, result 3 means alarm channel 1 and channel 2 have alarm now.

4.9.7 Get States of Alarm Output Channels

Table 4-88

Syntax	http://<server>/cgi-bin/alarm.cgi?action=getOutState
Method	GET
Description	Get alarm output state for all channels.
Example	http://192.168.1.250/cgi-bin/alarm.cgi?action=getOutState
Success	result=0

Return	
Comment	A bit in the response result indicates a channel, result 1 means alarm is present.

4.9.8 [Config] Video Blind Event

- Get video blind detect config

Table 4-89

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=BlindDetect
Method	GET
Description	Get blind detect config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=BlindDetect
Success Return	head.Enable=false head.EventHandler= (output of EventHandler is described in GetEventHandler) head.Level=3
Comment	Parameters in Response: head=table.BlindDetect[<i>ChannelNo</i>] ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

- Set video blind detect config

Table 4-90

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue> [&<paramName>=<paramValue> ...]
Method	GET
Description	Set blind detect config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&BlindDetect[0].Enable=true
Success Return	OK
Comment	Parameters in URL: In table below, head=BlindDetect[<i>ChannelNo</i>] ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

Appendix:

ParamName	ParamValue type	Description
head.Enable	bool	Enable/Disable blind detect feature.
head.EventHandler	—	Setting of EventHandler is described in SetEventHandler .
head.Level	integer	Range is [1—6]. Sensitivity of blind detection. 1: lowest sensitivity. 6: highest sensitivity.

4.9.9 [Config] Video Loss Event

- Get video loss detect config

Table 4-91

Syntax	<code>http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=LossDetect</code>
Method	GET
Description	Get video loss detect config.
Example	<code>http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=LossDetect</code>
Success Return	<p>head. Enable=false</p> <p>head. EventHandler= (output of EventHandler is described in GetEventHandler)</p>
Comment	<p>Parameters in Response:</p> <p>head=table.LossDetect [ChannelNo]</p> <p>ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p>

- Set video loss detect config

Table 4-92

Syntax	<code>http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]</code>
Method	GET
Description	Set video loss detection config.
Example	<code>http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&LossDetect[0].Enable=true</code>
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>In table below,</p> <p>head=LossDetect [ChannelNo]</p> <p>ChannelNo: integer, array index starts from 0, which means video channel>equals to video channel index -1, and so 0 means channel 1).</p>

Appendix:

ParamName	ParamValue type	Description
<code>head.Enable</code>	bool	Enable/Disable loss detect feature.
<code>head.EventHandler</code>	—	Setting of EventHandler is described in <u>SetEventHandler</u> .

4.9.10 [Config] Login Failure Event

- Get login failure event config

Table 4-93

Syntax	<code>http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=LoginFailureAlarm</code>
--------	--

Method	GET
Description	Get login failure alarm config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=LoginFailureAlarm
Success Return	<p>head.Enable=false</p> <p>head.EventHandler= (output of EventHandler is described in GetEventHandler)</p>

Comment Parameters in Response:
head=table.LoginFailureAlarm

- Set login failure alarm config

Table 4-94

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set login failure alarm config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&LoginFailureAlarm.Enable=true
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>In table below,</p> <p>head=LoginFailureAlarm</p>

Appendix:

ParamName	ParamValue type	Description
head.Enable	bool	Enable/Disable to notify LoginFailure event. Now this event can be linked with send email and alarm out. The max try login times can be configured in chapter <u>SetGeneralConfig</u> .
head.EventHandler	—	Setting of EventHandler is described in <u>SetEventHandler</u> .

4.9.11 [Config] Storage Not Exist Event

- Get storage not exist event config

Table 4-95

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=StorageNotExist		
Method	GET		
Description	Get storage not exist event config.		
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=StorageNotExist		
Success Return	<table border="0"> <tr> <td>table.StorageNotExist.Enable=false</td> </tr> <tr> <td>table.StorageNotExist.EventHandler= (output of EventHandler is described in GetEventHandler)</td> </tr> </table>	table.StorageNotExist.Enable=false	table.StorageNotExist.EventHandler= (output of EventHandler is described in GetEventHandler)
table.StorageNotExist.Enable=false			
table.StorageNotExist.EventHandler= (output of EventHandler is described in GetEventHandler)			

Comment	—
---------	---

- Set storage not exist event config

Table 4-96

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set storage not exist event config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&StorageNotExist.Enable=true
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
StorageNotExist.Enable	bool	Enable/Disable loss detect feature.
StorageNotExist.EventHandler	—	Setting of EventHandler is described in SetEventHandler .

4.9.12 [Config] Storage Access Failure Event

- Get storage access failure event config

Table 4-97

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=StorageFailure
Method	GET
Description	Get storage failure event config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=StorageFailure
Success Return	table.StorageFailure.Enable=false table.StorageFailure.EventHandler= (output of EventHandler is described in GetEventHandler)
Comment	—

- Set storage access failure event config

Table 4-98

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set storage failure event config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&StorageFailure.Enable=true

Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
StorageFailure.Enable	bool	Enable/Disable loss detect feature.
StorageFailure.EventHandler	—	Setting of EventHandler is described in <u>SetEventHandler</u> .

4.9.13 [Config] Storage Low Space Event

- Get storage low space event config

Table 4-99

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=StorageLowSpace
Method	GET
Description	Get storage low space event config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=StorageLowSpace
Success Return	table.StorageLowSpace.Enable=false table.StorageLowSpace.EventHandler= (output of EventHandler is described in GetEventHandler)
Comment	—

- Set storage low space event config

Table 4-100

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set storage low space event config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&StorageLowSpace.Enable=true
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
StorageLowSpace.Enable	bool	Enable/Disable loss detect feature.
StorageLowSpace.EventHandler	—	Setting of EventHandler is described in <u>SetEventHandler</u> .

4.9.14 [Config] Net Abort Event

- Get net abort event config

Table 4-101

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=NetAbort
Method	GET
Description	Get net abort event config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=NetAbort
Success Return	table.NetAbort.Enable=false table.NetAbort.EventHandler= (output of EventHandler is described in GetEventHandler)
Comment	—

- Set net abort event config

Table 4-102

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set net abort event config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&NetAbort.Enable=true
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
NetAbort.Enable	bool	Enable/Disable loss detect feature.
NetAbort.EventHandler	—	Setting of EventHandler is described in SetEventHandler .

4.9.15 [Config] IP Conflict Event

- Get IP conflict event config

Table 4-103

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=IPConflict
Method	GET
Description	Get IP conflict event config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=IPConflict
Success Return	table.IPConflict.Enable=false table.IPConflict.EventHandler= (output of EventHandler is described in GetEventHandler)

Comment	—
<ul style="list-style-type: none"> • Set IP conflict event config 	
Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set IP Conflict event config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&IPConflict.Enable=true
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
IPConflict.Enable	bool	Enable/Disable loss detect feature.
IPConflict.EventHandler	—	Setting of EventHandler is described in SetEventHandler .

4.9.16 Get Channels Event Happened

Table 4-105

Syntax	http://<server>/cgi-bin/eventManager.cgi?action=getEventIndexes&code=<eventCode>
Method	GET
Description	Get channels indexes that event of code eventCode happens. Not all events support this command. Do not recommend to use it, use Attach command instead.
Example	http://192.168.1.250/cgi-bin/eventManager.cgi?action=getEventIndexes&code=AlarmLocal
Success Return	channels[0]=0 channels[1]=2 channels[2]=3 ... (This response means event happened on channel 0, channel 2 and channel 3 while video channel index starts from 0)
Comment	Parameters in URL: eventCode includes: VideoMotion: motion detection event VideoLoss: video loss detection event VideoBlind: video blind detection event. AlarmLocal: alarm detection event. StorageNotExist: storage not exist event. StorageFailure: storage failure event.

StorageLowSpace: storage low space event.

AlarmOutput: alarm output event.

4.9.17 Subscribe to Event Message

Subscribe to event messages. For details on the event name and parameters, see the corresponding [Event] sections.

Request URL	http://<server>/cgi-bin/eventManager.cgi?action=attach			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	
codes	string	R	<p>Event code, can be a list of event codes or "All" to cover all event codes.</p> <p>Here are some frequently used event codes:</p> <ul style="list-style-type: none">VideoMotion: motion detection eventSmartMotionHuman: human smart motion detectionSmartMotionVehicle: Vehicle smart motion detectionVideoLoss: video loss detection eventVideoBlind: video blind detection event.AlarmLocal: alarm detection event.StorageNotExist: storage not exist event.StorageFailure: storage failure event.StorageLowSpace: storage low space event.AlarmOutput: alarm output event.AudioMutation: intensity changeAudioAnomaly: input abnormalCrossLineDetection: tripwire eventCrossRegionDetection: intrusion eventLeftDetection: abandoned object detectionTakenAwayDetection: missing object detectionVideoAbnormalDetection: scene change eventFaceDetection: face detect eventVideoUnFocus: defocus detect eventWanderDetection: loitering detection eventRioterDetection: People Gathering eventParkingDetection: parking detection eventMoveDetection: fast moving eventHeatImagingTemper: temperature alarm eventCrowdDetection: crowd density overrun eventFireWarning: fire warning eventFireWarningInfo: fire warning specific data infoObjectPlacementDetection: placement object detectionObjectRemovalDetection: removal object detection	Example [AlarmLocal, VideoMotion, VideoLoss, VideoBlind, FaceDetection]

			AccessControl: Access control event	
keepalive	int	O	<p>Client keep-alive. If this parameter exists, the client will send keep-alive data to the device at an interval in seconds; the value range is [1–60].</p> <p>The keep-alive data can be "keep alive" in the form of string.</p> <p>Note: It is recommended to use the Heartbeat parameter instead of the keepalive parameter.</p>	20
heartbeat	int	O	<p>Server keep-alive; integer; Unit: s;</p> <p>The value range is [1–60].</p> <p>For example, if the parameter exists in the URL and the value is 5, the device will send "heartbeat" to the client as a keep-alive message.</p> <p>Note: The keep-alive message must be sent before the keepalive parameter expires.</p>	5

Request Example

```
http://192.168.1.250/cgi-bin/eventManager.cgi?action=attach&codes=[AlarmLocal%2CVideoMotion%2CVideoLoss%2CVideoBlind%2CFaceDetection]
```

Response Params (multipart , key=value format in body , Heartbeat in body)

Name	Type	R/O	Description	Example
Code	string	R	Event code	VideoBlind
action	string	R	Event action Values: Start/Stop/pulse	Start
index	int	R	Number; meaningless sometimes.	0
data	object	O	Event data; json message body. For detailed message bodies, see the corresponding section.	""

Response Example

HTTP/1.1 200 OK

Cache-Control: no-cache

Pragma: no-cache

Expires: Thu, 01 Dec 2099 16:00:00 GMT

Connection: close

Content-Type: multipart/x-mixed-replace; boundary=myboundary

--myboundary

Content-Type: text/plain

Content-Length: 39

Code=VideoMotion;action=Start;index=0

```
--myboundary
Content-Type: text/plain
Content-Length: 38

Code=FaceDetection;action=Start;index=0;data={
  "Faces": [ { "BoundingBox": [2992,136,6960,8192],
    "Sex": "Man",
    "Age": 40,
    "Feature": [ "WearGlasses", "Smile" ],
    "Eye": 2,
    "Mouth": 1,
    "Mask": 1,
    "Beard": 2
  }, {...}, ... ]
}
```

```
--myboundary
Content-Type: text/plain
Content-Length: 9
```

Heartbeat

```
--myboundary
.....
```

4.9.18 Get Capability of Event Management

Table 4-106

Syntax	<a href="http://<server>/cgi-bin/eventManager.cgi?action=getCaps">http://<server>/cgi-bin/eventManager.cgi?action=getCaps
Method	GET
Description	Get event manager capabilities.
Example	http://192.168.1.250/cgi-bin/eventManager.cgi?action=getCaps
Success Return	caps.AlarmOutEnable=true caps.BeepEnable=true caps.DejitterEnable=true caps.MMSEnable=true caps.MailEnable=true caps.MonitorTourEnable=true caps.PtzLinkEnable=true caps.RecordEnable=true caps.SnapshotEnable=true caps.TimeSectionEnable=true caps.TipEnable=true caps.RecordCloudEnable=true, caps.SnapshotCloudEnable=false, caps.SnapshotTimes[0]=3,

	<pre> caps.SnapshotTimes[1]=10, caps.SupportAlarmBell=false, caps.SupportAccessControl=false, caps.SipCallEnable=false, caps.SupportAlarmServer=false, caps.SupportPtzLinkDelay=false, caps.SupportPSTNAlarmServer=false, caps.SupportICR=false, caps.BeepTime=false, caps.DejitterRange.Min=0, caps.DejitterRange.Max=60 caps.AlarmOutLatch[0]=1, caps.AlarmOutLatch[1]=300, caps.RecordLatch[0]=0, caps.RecordLatch[1]=300, caps.VoiceEnable=true, caps.VoiceLinkTimeRange[0]=10, caps.VoiceLinkTimeRange[1]=30, caps.VoicePlayTimesRange[0]=1, caps.VoicePlayTimesRange[1]=10, caps.VoiceLinkFileOptional=true, caps.LogEnable=true, caps.SupportLightControl=false, caps.LinkLightBrightRange[0]=0, caps.LinkLightBrightRange[1]=100, caps.SupportVideoMatrix=false, caps.LinkDetailCameraEnable=false, caps.SupportDisableLinkage[0]=0 caps.SupportDisableLinkage[1]=0 caps.SupportDisableLinkage[2]=0 caps.SupportDisableLinkage[3]=0 caps.SupportHTTPUpload.PictureHttpUploadEnable=false caps.SupportHTTPUpload.EventHttpUploadEnable=false </pre>
Comment	—

4.9.19 [Config] Net Alarm Event

Net alarm event configuration parameters:

Config Data Params				
Name	Type	R/O	Description	Example
NetAlarm	object[]	R	Network alarm; one-dimensional array. The array index represents the corresponding channel which starts from 0.	
+Enable	bool	O	Enable alarm input (bypass switch)	true
+DefenceAreaType	string	O	Type of the protection zone enumchar[32]{	"Intime"

			<p>"Intime": Real-time protection zone "Delay": Time-delay protection zone "Fullday": 24-hour protection zone Get the capability set to judge the protection zone type. }</p>	
+DisableDelay	integer	O	<p>Delay disarming time Unit: s Get the maximum delay time from the capability set. Effective when EnableAlways is selected or when the EnableControl is set to Normal. Others: Effective only when the protection zone type is Delay.</p>	30
+EnableDelay	integer	O	<p>Delay arming time Unit: s Get the maximum delay time from the capability set. Effective when the protection zone type is Delay.</p>	30
+Name	string	O	Alarm Channel Name	"Door"
+SensorType	string	O	<p>The sensor is solid on or off. enumchar[32]{ "NC" "NO" }</p>	NC
+EventHandler	object	O	Alarm linkage <p>For linkage configurations, see SetEventHandler.</p>	

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=NetAlarm
```

Get Config Response Example

```
table.NetAlarm[0].Enable=false
table.NetAlarm[0].Name=channel1
table.NetAlarm[0].SensorType=NO
table.NetAlarm[0].EventHandler= (output of EventHandler is described in GetEventHandler)
```

Set Config Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&NetAlarm[0].Enable=true
```

Set Config Response Example

```
OK
```

4.9.20 Set Net Alarm State

Set the network alarm status.

Request URL	http://<server>/cgi-bin/netAlarm.cgi?action=setState		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	R	Channel number; access control number; starting from 1
alarm	bool	R	Alarm status; bool; "true" means the alarm is enabled
name	string	O	Alarm name; string; less than 32 characters
trigger	string	O	Alarm triggering source; string; less than 32 characters. For example, Infrared, WaterSensor Manual and more
desc	string	O	Alarm description; string; less than 128 characters
Request Example			
http://192.168.1.250/cgi-bin/netAlarm.cgi?action=setState&channel=1&alarm=true&name=somke&trigger=SmokingSensor&desc=Zone8			

Response Params (OK in body)
Response Example
OK

4.9.21 Get Supported Events

Get the event list which the device supports.

Request URL	http://<server>/cgi-bin/eventManager.cgi?action=getExposureEvents					
Method	GET					
Request Params (none)						
Request Example						
http://192.168.1.250/cgi-bin/eventManager.cgi?action=getExposureEvents						

Response Params (key=value format in body)
Response Example
events[0]=VideoMotion
events[1]=AlarmLocal
events[2]=FaceDetection
events[3]=VideoMotion
...

4.10 Record

4.10.1 Get Capability of Recording

Table 4-107

Syntax	http://<server>/cgi-bin/recordManager.cgi?action=getCaps
Method	GET
Description	Get record manager capabilities.
Example	http://192.168.1.250/cgi-bin/recordManager.cgi?action=getCaps
Success Return	caps.MaxPreRecordTime=30 caps.PacketLengthRange[0]=1 caps.PacketLengthRange[1]=60 caps.PacketSizeRange[0]=131072 caps.PacketSizeRange[1]=2097152 caps.SupportExtraRecordMode=true caps.SupportHoliday=true caps.SupportPacketType[0]=Time caps.SupportPacketType[1]=Size caps.SupportResumeTransmit=false
Comment	—

4.10.2 [Config] Record Config

- Get record config

Table 4-108

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=Record
Method	GET
Description	Get record config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=Record
Success Return	table.Record[channel].PreRecord=6 table.Record[channel].TimeSection[weekday][0]=65535 00:00:00-24:00:00 table.Record[channel].TimeSection[weekday][1]=0 02:00:00-24:00:00 table.Record[channel].TimeSection[weekday][2]=0 03:00:00-24:00:00 table.Record[channel].TimeSection[weekday][3]=0 04:00:00-24:00:00 table.Record[channel].TimeSection[weekday][4]=0 05:00:00-24:00:00 table.Record[channel].TimeSection[weekday][5]=0 06:00:00-24:00:00
Comment	Parameters in Response: channel : integer, array index starts from 0, which means video channel>equals to video channel index -1, and so 0 means channel 1). weekday : range is [0-6] (Sunday - Saturday). Record config contains pre record time and record time sections of every day.

- Set record config

Table 4-109

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set record config.
Example	<p>http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&Record[0].TimeSection[0][0]=6 00:00:00-23:59:59</p> <p>Set record time to every Sunday all day. Record type is motion detection and alarm.</p> <p>In this example, "6 00:00:00-23:59:59" means motion detection and alarm record all day (6 = 4 & 2, alarm is 4, motion detection is 2.).</p>
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>In table below,</p> <p>ch: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p> <p>wd: week day index</p> <p>ts: time section index</p>

Appendix:

ParamName	ParamValue type	Description
Record[ch].PreRecord	integer	<p>Range is [0—300].</p> <p>Prerecord seconds, 0 means no prerecord.</p> <p>ch (Channel number) starts form 0</p>
Record[ch].TimeSection[wd][ts]	string	<p>wd (week day) range is [0—7] ,[0-6]: (Sunday - Saturday), 7: Holiday</p> <p>ts (time section) range is [0—23], time section table index.</p> <p>Format: mask hh:mm:ss-hh:mm:ss</p> <p>Mask: [0—4294967295], hh: [0—24], mm: [0—59], ss: [0—59]</p> <p>Mask indicates record type by bits:</p> <p>Bit0: regular record</p> <p>Bit1: motion detection record</p> <p>Bit2: alarm record</p> <p>Bit3: card record</p>

4.10.3 [Config] Record Mode

- Get record mode config

Table 4-110

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=RecordMode
Method	GET

Description	Get record mode config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=RecordMode
Success Return	table.RecordMode[Channel].Mode=0
Comment	<p>Parameters in Response:</p> <p>Channel: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p>

- Set record mode config

Table 4-111

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&< paramName >=< paramValue >
Method	GET
Description	Set record mode config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&RecordMode[0].Mode=0
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>In table below,</p> <p>Channel: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p>

Appendix:

ParamName	ParamValue type	Description
RecordMode[Channel].Mode	integer	<p>Range is {0, 1, 2}.</p> <p>0: automatically record</p> <p>1: manually record</p> <p>2: stop record.</p>

4.10.4 [Config] Media Global

- Get media global config

Table 4-112

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=MediaGlobal
Method	GET
Description	Get media global config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=MediaGlobal
Success Return	table.MediaGlobal.OverWrite=true, table.MediaGlobal.PacketType=0, table.MediaGlobal.PacketLength=60, table.MediaGlobal.PacketSize=1024, table.MediaGlobal.LogRecord=false, table.MediaGlobal.LogEncode=false,

	table.MediaGlobal.SnapFormatAs>MainFormat
Comment	—

- Set media global config

Table 4-113

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>
Method	GET
Description	Set media global config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&MediaGlobal.SnapFormatAs>MainFormat
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
MediaGlobal.SnapFormatAs	string	The range is {"MainFormat", "ExtraFormat"}.

4.10.5 Find Media Files

Step 1 Create a media files finder.

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=factory.create					
Method	GET					
Request Params (none)						
Request Example						
http://192.168.1.250/cgi-bin/mediaFileFind.cgi?action=factory.create						

Response Params (key=value format in body)				
Name	Type	R/O	Description	Sample
result	string	R	objectid; use it to search for media files	08137
Response Example				
result=08137				

Step 2 Start to find media files satisfied the conditions with the finder.

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findFile					
Method	GET					
Request Params (key=value format in URL)						
Name						
object	Type	R/O	Description	Sample		
	string	R	Search object of the media files; Create the object in the previous command, and then search for the object.	08137		
condition	object	R	user id	101		
+Channel	int	R	Channel number; starting from 1	1		

+StartTime	string	R	Start time	"2010-05-25 12:05:00"
+EndTime	string	R	End time	"2010-05-25 12:10:00"
+Dirs	string[]	O	The directory where the recordings are saved; array; the index starts from 0; optional values: {"dav", "jpg", "mp4"} If this parameter does not exist, the system will search for all directories.	["/mnt/dvr/sda0", "/mnt/dvr/sda1"]
+Types	string[]	O	The type of file that is being searched for; array; the index starts from 0; optional values: {"dav", "jpg", "mp4"} If this parameter does not exist, the system will search for all file types.	["dav"]
+Flags	string[]	O	Search parameter; array; the index starts from 0; optional values: {"Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout"} If this parameter does not exist, the system will search for all files.	["Timing"]
+Events	string[]	O	Events related to the recordings; array; the index starts from 0; optional values: {"AlarmLocal", "VideoMotion", "VideoLoss", "VideoBlind", "Traffic*", ...} If this parameter does not exist, the system will search for all files.	["AlarmLocal"]
+VideoStream	string	O	Stream type; optional values: {"Main", "Extra1", "Extra2", "Extra3"}. If this parameter does not exist, the system will search for all stream types.	Main

Request Example

```
http://192.168.1.250/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.Dirs[0]=/mnt/dvr/sda0&condition.Types[0]=dav&condition.Events[0]=AlarmLocal&condition.Events[1]=VideoMotion&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.VideoStream=Main
```

Response Params (OK in body)

Response Example

OK

Step 3 Get the media file information found by the finder.

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findNextFile			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample
object	string	R	The search object of the media files; Create the object in the previous command,	08137

			and then search for the object.	
count	int	R	Get the number of files that have been found, and the maximum number is 100.	100

Request Example

<http://192.168.1.250/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100>

Response Params (key=value format in body)

Name	Type	R/O	Description	Sample
found	int	O	The returned number of files that have been found. 0 means no file has been found.	1
items	object[]	O	List of file information	
+Channel	int	O	Channel number; integer; the video channel number starts from 0.	1
+StartTime	string	O	Start time of the recording	2011-1-1 12:00:00
+EndTime	string	O	End time of the recording	2011-1-1 13:00:00
+Type	string	O	File type	dav
+Events	string[]	O	Type of events that are related to the recordings	["AlarmLocal"]
+VideoStream	string	O	Stream type	Main
+FilePath	string	O	File path	/mnt/dvr/sda0/2010/8/1 1/dav/15:40:50.jpg
+Length	int	O	The file length within a specified period	792
+Duration	int	O	File length	3600

Response Example

```
found=1
items[0].Channel=1
items[0].StartTime=2011-1-1 12:00:00
items[0].EndTime=2011-1-1 13:00:00
items[0].Type=dav
items[0].Events[0]=AlarmLocal
items[0].VideoStream=Main
items[0].FilePath=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg
items[0].Length=790
items[0].Duration=3600
```

Step 4 Close the finder.

Request URL	<a href="http://<server>/cgi-bin/mediaFileFind.cgi?action=close">http://<server>/cgi-bin/mediaFileFind.cgi?action=close			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample
object	string	R	The search object of the media files; Create the object in the previous command, and then search for the object.	08137

Request Example

<http://192.168.1.250/cgi-bin/mediaFileFind.cgi?action=close&object=08137>

Response Params (OK in body)**Response Example**

OK

Step 5 Destroy the finder.

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=destroy		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
object	string	R	The search object of the media files; Create the object in the previous command, and then search for the object.
Request Example			
http://192.168.1.250/cgi-bin/mediaFileFind.cgi?action=destroy&object=08137			

Response Params (OK in body)**Response Example**

OK

4.10.6 Find media files with FaceDetection info

Step 1 Create a media files finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and FaceDetection condition with the finder.

Table 4-114

URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findFile		
Method	GET		
Description	Start to find media files satisfied the common conditions and FaceDetection conditions with the finder.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
condition	object	R	The search conditions
+Channel	int	R	The video channel to search, video channel index start from 1, use -1 to search all video channel.
+StartTime	string	R	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	R	End time to search, ex: "2010-06-25 12:05:00"
+Types	array<string>	O	Search file types, should be "jpg"
+Flags	array<string>	O	Search flags, can be : "Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout" should include "Event"

+Events	array<string>	R	Search event list, must be one string : " FaceDetection "
+Dirs	array<string>	O	Search directory list, if omit, search all. Each dir path max string length is 259.
+DB	object	R	DB filter object
++FaceDetecti onRecordFilter	object	R	DB filter for FaceDetection
+++ImageType	string	O	Picture type, if omit, means search all type. "GlobalSence" : means big picture of the full sence, "Small" : means small picture of the people face.
+++Sex	string	O	Sex, it can be "Man", "Woman", if omit, search all
+++Age	array<int>	O	Age range, ex : [25, 40]
+++Glasses	int	O	Glasses Status, 0: all, 1: not wear, 2: wear
+++Mask	int	O	Mask Status, 0: all, 1: not wearing mask, 2: wearing mask
+++Beard	int	O	Beard Status, 0: all, 1: no beard, 2: has beard
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.250/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[0]=jpg&condition.Flags[0]=Event&condition.Events[0]=FaceDetection&condition.DB.FaceDetectionRecordFilter.ImageType=GlobalSence&condition.DB.FaceDetectionRecordFilter.Sex=Man&condition.DB.FaceDetectionRecordFilter.Age[0]=25&condition.DB.FaceDetectionRecordFilter.Age[1]=40&condition.DB.FaceDetectionRecordFilter.Glasses=1		
Response	OK		

Step 3 Get the media file information found by the finder

Table 4-115

URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findNextFile		
Method	GET		
Description	Get the media file information found by the finder.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
count	int	R	Record num to get, should less than 100.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
found	int	R	Record num that found.
items	array<object>	R	Result record items.
+Channel	int	O	The video channel index start from 0,
+StartTime	string	R	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	R	End time to search, ex: "2010-06-25 12:05:00"
+Type	string	O	File type, should be: "jpg"
+Events	array<string>	O	Events with this record

+FilePath	string	O	File path, string max length is 259
+CutLength	int	O	File length that cut between start time and end time
+Length	int	O	File length of the whole file
+SummaryNew	array<object>	R	DB record object
++Key	string	R	DB record name, should be " FaceDetectionRecord "
++Value	object	R	DB record value
+++ImageType	string	O	Picture type. It can be: "GlobalSence" , "Small"
+++TimeStamp	object	O	Picture timestamp
++++UTC	int	O	UTC seconds
++++UTCMS	int	O	UTC miliseconds
+++Sex	string	O	Sex, it can be "Man", "Woman", "Unknown"
+++Age	int	O	Age
+++Glasses	int	O	Glasses Status, 0: unknown, 1: not wear, 2: wear
+++Mask	int	O	Mask Status, 0: unknown, 1: not wearing mask, 2: wearing mask
+++Beard	int	O	Beard Status, 0: unknown, 1: no beard, 2: has beard

[Example]

Request	GET http://192.168.1.250/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100
Response	found=100 items[0].Channel=1 items[0].StartTime=2011-1-1 12:00:00 items[0].EndTime=2011-1-1 13:00:00 items[0].Type=jpg items[0].Events[0]=FaceDetection items[0].FilePath=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg items[0].CutLength=79000 items[0].SummaryNew[0].Key=FaceDetectionRecord items[0].SummaryNew[0].Value.ImageType=GlobalSence items[0].SummaryNew[0].Value.TimeStamp.UTC=134652732 items[0].SummaryNew[0].Value.TimeStamp.UTCMS=134 items[0].SummaryNew[0].Value.Sex=Man items[0].SummaryNew[0].Value.Age=30 items[0].SummaryNew[0].Value.Glasses=1 items[0].SummaryNew[0].Value.Mask=2 items[0].SummaryNew[0].Value.Beard=1 ...

Step 4 Close the finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.10.5 Find Media Files".

4.10.7 Find media files with Face Recognition info

Step 1 Create a media files finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and FaceRecognition conditions with the finder.

Table 4-116

URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findFile		
Method	GET		
Description	Start to find media files satisfied the common conditions and FaceRecognition conditions with the finder.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
condition	object	R	The search conditions
+Channel	int	R	The video channel to search, video channel index start from 1, use -1 to search all video channel.
+StartTime	string	O	Start time to search, ex: "2010-05-25 12:05:00" Ignored, use StartTime in DB param
+EndTime	string	O	End time to search, ex: "2010-06-25 12:05:00" Ignored, use EndTime in DB param
+Types	array<string>	O	Search file types, should be "jpg"
+Flags	array<string>	O	Search flags. It can be: "Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout" should include "Event".
+Events	array<string>	R	Search event list, must be one string: "FaceRecognition"
+Dirs	array<string>	O	Search directory list, if omit, search all. Each dir path max string length is 259.
+DB	object	R	DB filter object
++FaceRecognitionRecordFilter	object	R	DB filter for FaceRecognition
+++MachineAddress	string	O	Machine address
+++StartTime	string	R	Start time to search, ex: "2010-05-25 12:05:00"
+++EndTime	string	R	End time to search, ex: "2010-06-25 12:05:00"
+++Person	object	O	Search condition of history person and the similar face group person candidates
++++Name	string	O	Person's name, max string length is 15
++++Sex	string	O	Sex, can be "Male", "Female", if omit, search all
++++Birthday	string	O	Birthday, max string length is 11, ex: 1990-5-1
++++Country	string	O	Country, ISO 3166, string length should be 2
++++Province	string	O	Province, max string length is 63

++++City	string	O	City, max string length is 63
++++Certificate Type	string	O	Certificate Type. It can be: "IC", "Passport", "Unknown"
++++ID	string	O	Person ID of CertificateType, max string length is 31
++++GroupID	string	O	The identity of the Face Group that this Person in. max string length is 63
++++Age	array<int>	O	Age range, ex : [25, 40]
++++Glasses	int	O	Glasses Status, 0: all, 1: not wear, 2: wear
++++Mask	int	O	Mask Status, 0: all, 1: not wearing mask, 2: wearing mask
++++Beard	int	O	Beard Status, 0: all, 1: no beard, 2: has beard
+++GroupID	array<string>	O	GroupID list
++SimilaryRange	array<int>	O	Similary Range, ex: [40, 100]
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.250/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[0]=jpg&condition.Flags[0]=Event&condition.Events[0]=FaceRecognition&condition.DB.FaceRecognitionRecordFilter.RegType=RecSuccess&condition.DB.FaceRecognitionRecordFilter.StartTime=2014-1-1%2012:00:00&condition.DB.FaceRecognitionRecordFilter.EndTime=2015-1-10%2012:00:00&condition.DB.FaceRecognitionRecordFilter.Person.Sex=Male&condition.DB.FaceRecognitionRecordFilter.Person.Country=IND&condition.DB.FaceRecognitionRecordFilter.Person.Age[0]=25&condition.DB.FaceRecognitionRecordFilter.Person.Age[1]=40&condition.DB.FaceRecognitionRecordFilter.Person.Glasses=1&condition.DB.FaceRecognitionRecordFilter.GroupID[0]=10001&condition.DB.FaceRecognitionRecordFilter.GroupID[1]=10003&condition.DB.FaceRecognitionRecordFilter.GroupID[2]=10005&condition.DB.FaceRecognitionRecordFilter.SimilaryRange[0]=40&condition.DB.FaceRecognitionRecordFilter.SimilaryRange[1]=100		
Response	OK		

Step 3 Get the media file information found by the finder

Table 4-117

URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findNextFile		
Method	GET		
Description	Get the media file information found by the finder		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
count	int	R	Record num to get, should less than 100.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
found	int	R	Record num that found.
items	array<object>	R	Result record items.

+Channel	int	O	The video channel index start from 0,
+StartTime	string	R	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	R	End time to search, ex: "2010-06-25 12:05:00"
+Type	string	O	File type, should be: "jpg"
+Events	array<string>	O	Events with this record
+FilePath	string	O	File path, string max length is 259
+CutLength	int	O	File length that cut between start time and end time
+Length	int	O	File length of the whole file
+SummaryNew	array<object>	R	DB record object
++Key	string	R	DB record name, should be " FaceRecognitionRecord "
++Value	object	R	DB record value
+++RecResult	int	O	Recognition result, 0 means recognition failed, no candidates. 1 means recognition success, has candidates.
+++MachineAddress	string	O	Machine address, string max length is 259
+++IsGlobalScene	bool	O	Is the global scene picture or not
+++ImageInfo	object	O	Big picture info
++++Length	int	O	The length of the picture
++++FilePath	string	O	The file path of the picture, max string length is 259
+++Object	object	O	The target face info
++++Sex	string	O	Sex, it can be "Man", "Woman", "Unknown"
++++Age	int	O	Age
++++Glasses	int	O	Glasses Status, 0: unknown, 1: not wear, 2: wear
++++Eye	int	O	Eye status, 0: not detected, 1: close eye, 2: open eye
++++Mouth	int	O	Mouth status, 0: not detected, 1: close mouth, 2: open mouth
++++Mask	int	O	Mask status, 0: not detected, 1: not wearing mask, 2: wearing mask
++++Beard	int	O	Beard status, 0: not detected, 1: no beard, 2: has beard
+++Candidates	array<object>	O	The info of candidates face from face group.
++++Similarity	int	O	Similarity
++++Person	object	O	person info
++++Name	string	O	Person Name, max string length is 63
++++Sex	string	O	Sex, it can be "Male", "Female", "Unknown"
++++Birthday	string	O	The person's birthday, ex: "1980-01-01"
++++Country	string	O	Country name, length must be 2, value should be according to ISO3166
++++Province	string	O	Province name, max string length is 63
++++City	string	O	City name, max string length is 63
++++CertificateType	string	O	Certificate Type. It can be: "IC", "Passport", "Unknown"
++++ID	string	O	Person ID of CertificateType, max string length is 31
++++FeatureState	int	O	Feature State, 0:Unknown, 1:Failed, 2:OK
[Example]			
Request	GET		

	http://192.168.1.250/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100
Response	<pre> found=100 items[0].Channel=1 items[0].StartTime=2011-1-1 12:00:00 items[0].EndTime=2011-1-1 13:00:00 items[0].Type=jpg items[0].Events[0]=FaceRecognition items[0].FilePath =/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg items[0].Length =790 items[0].SummaryNew[0].Key=FaceRecognitionRecord items[0].SummaryNew[0].Value.RecResult=1 items[0].SummaryNew[0].Value.MachineAddress=AAA items[0].SummaryNew[0].Value.IsGlobalScene=true items[0].SummaryNew[0].Value.ImageInfo.Length=123 items[0].SummaryNew[0].Value.ImageInfo.FilePath=/tmp/1.jpg items[0].SummaryNew[0].Value.Object.Sex=Man items[0].SummaryNew[0].Value.Object.Age=40 items[0].SummaryNew[0].Value.Object.Glasses=1 items[0].SummaryNew[0].Value.Object.Eye=2 items[0].SummaryNew[0].Value.Object.Mouth=1 items[0].SummaryNew[0].Value.Candidates[0].Similarity=50 items[0].SummaryNew[0].Value.Candidates[0].Person.Name=Rohit items[0].SummaryNew[0].Value.Candidates[0].Person.Birthday=1980-01-01 items[0].SummaryNew[0].Value.Candidates[0].Person.Sex=Male items[0].SummaryNew[0].Value.Candidates[0].Person.Country=IND items[0].SummaryNew[0].Value.Candidates[0].Person.Province=XXX items[0].SummaryNew[0].Value.Candidates[0].Person.City=YYY items[0].SummaryNew[0].Value.Candidates[0].Person.CertificateType=IC items[0].SummaryNew[0].Value.Candidates[0].Person.ID=1234567890 items[0].SummaryNew[0].Value.Candidates[0].Person.FeatureState=0 ... </pre>

Step 4 Close the finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.10.5 Find Media Files".

4.10.8 Find media files with HumanTrait info

Step 1 Create a media files finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and HumanTrait conditions with the finder

Table 4-118

URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findFile		
Method	GET		
Description	Start to find media files satisfied the common conditions and HumanTrait conditions with the finder.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
condition	object	R	The search conditions
+Channel	int	R	The video channel to search, video channel index start from 1, use -1 to search all video channel.
+StartTime	string	O	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	O	End time to search, ex: "2010-06-25 12:05:00"
+Types	array<string>	O	Search file types, should be "jpg"
+Flags	array<string>	O	Search flags. It can be: "Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout" should include "Event"
+Events	array<string>	R	Search event list, must be one string: "HumanTrait"
+Dirs	array<string>	O	Search directory list, if omit, search all. Each dir path max string length is 259.
+DB	object	R	DB filter object
++HumanTrait RecordFilter	object	R	DB filter for HumanTrait
+++HumanAttributes	object	R	Human attributes
++++CoatColor	array<string>	O	Coat color, refer to CoatColor in 16.1.3.5 [Event] HumanTrait for available values, max array size is 4
++++CoatType	array<int>	O	Coat type, refer to CoatType in 16.1.3.5 [Event] HumanTrait for available values, max array size is 2
++++Trousers Color	array<string>	O	Trousers color, refer to TrousersColor in 16.1.3.5 [Event] HumanTrait for available values, max array size is 4
++++TrousersT ype	array<int>	O	Trousers type, refer to TrousersType in 16.1.3.5 [Event] HumanTrait for available values, max array size is 2
++++HasHat	int	O	Has hat or not, 0: all, 1: not has hat, 2: has hat
++++HasBag	int	O	Has bag or not, 0: all, 1: not has bag, 2: has bag
++++Sex	string	O	Sex, can be "Man", "Woman", if omit, search all
++++Age	array<int>	O	Age range, ex : [25, 40]
++++HairStyle	int	O	Hair style, 0: all, 1: long hair, 2: short hair, 3: ponytail, 4: updo, 5: hiddened
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.250/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[0]=jpg&condition.Flags[0]=Event&condition.Events[0]=HumanTrait		

	t&condition.DB.HumanTraitRecordFilter.HumanAttributes.CoatColor[0]=White&condition.D B.HumanTraitRecordFilter.HumanAttributes.CoatColor[1]=Yellow&condition.DB.HumanTrait RecordFilter.HumanAttributes.CoatType=1&condition.DB.HumanTraitRecordFilter.HumanAt tributes.HasHat=2&condition.DB.HumanTraitRecordFilter.HumanAttributes.Sex=Man&cond ition.DB.HumanTraitRecordFilter.HumanAttributes.Age[0]=30&condition.DB.HumanTraitRe cordFilter.HumanAttributes.Age[1]=50&condition.DB.HumanTraitRecordFilter.HumanAttribu tes.HairStyle=1
Respon se	OK

Step 3 Get the media file information found by the finder

Table 4-119

URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findNextFile		
Method	GET		
Description	Get the media file information found by the finder		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
count	int	R	Record num to get, should less than 100.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
found	int	R	Record num that found.
items	array<object>	R	Result record items.
+Channel	int	O	The video channel index start from 0,
+StartTime	string	R	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	R	End time to search, ex: "2010-06-25 12:05:00"
+Type	string	O	File type, should be: "jpg"
+Events	array<string>	O	Events with this record
+FilePath	string	O	File path, string max length is 259
+CutLength	int	O	File length that cut between start time and end time
+Length	int	O	File length of the whole file
+SummaryNew	object	R	Face info
++Key	string	R	DB record name, should be " HumanTraitRecord "
++Value	object	R	DB record value
+++HumanAttri butes	object	O	Human attributes
++++CoatColor	string	O	Coat color, refer to CoatColor in 16.1.3.5 [Event] HumanTrait for available values,
++++CoatType	int	O	Coat type, refer to CoatType in 16.1.3.5 [Event] HumanTrait for available values,
++++Trousers Color	string	O	Trousers color, refer to TrousersColor in 16.1.3.5 [Event] HumanTrait for available values,
++++TrousersT ype	int	O	Trousers type, refer to TrousersType in 16.1.3.5 [Event] HumanTrait for available values,

++++HasHat	int	O	Has hat or not, 0: unknown, 1: not has hat, 2: has hat
++++HasBag	int	O	Has bag or not, 0: unknown, 1: not has bag, 2: has bag
++++Sex	string	O	Sex, can be "Man", "Woman", "Unknown"
++++Age	int	O	Age
++++HairStyle	int	O	Hair style, 0: unknown, 1: long hair, 2: short hair, 3: ponytail, 4: updo, 5: hiddened
++++HasUmbr ella	int	O	Has umbrella or not, 0: unknown, 1: not has umbrella, 2: has umbrella
++++Bag	int	O	Bag type, 0: unknown, 1: handbag, 2: shoulder bag, 3: knapsack, 4: draw-bar box
++++Cap	int	O	Cap style, 0: unknown, 1: normal cap, 2: helmet
+++FaceAttribu tes	object	O	Face attributes
++++Sex	string	O	Sex, can be "Man", "Woman", "Unknown"
++++Age	int	O	Age
++++Mask	int	O	Mask status, 0: not detected, 1: not wearing mask, 2: wearing mask
++++Beard	int	O	Beard status, 0: not detected, 1: no beard, 2: has beard
++++Glass	Int	O	Glasses Status, 0: unknown, 1: not wearing, 2: normal Glasses, 3: sun glasses, 4: black frame glasses
++++Emotion	string	O	Emotion info. It can be: "Unknown", "Smile", "Anger", "Sadness", "Disgust", "Fear", "Surprise", "Neutral", "Laugh", "Happy", "Confused", "Scream", "Lookaside"
+++FacePath	string	O	Face picture path, max string length is 259
+++FaceScene Path	string	O	Face scene picture path, max string length is 259
+++HumanPat h	string	O	Human picture path, max string length is 259
+++HumanSce nePath	string	O	Human scene picture path, max string length is 259
[Example]			
Request	GET http://192.168.1.250/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100		

	<pre> found=100 items[0].Channel=1 items[0].StartTime=2011-1-1 12:00:00 items[0].EndTime=2011-1-1 13:00:00 items[0].Type=jpg items[0].Events[0]=HumanTrait items[0].FilePath=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg items[0].Length=790 items[0].SummaryNew.Key=HumanTraitRecord items[0].SummaryNew.Value.HumanAttributes.CoatColor=White items[0].SummaryNew.Value.HumanAttributes.CoatType=1 items[0].SummaryNew.Value.HumanAttributes.TrousersColor=Black items[0].SummaryNew.Value.HumanAttributes.TrousersType=1 items[0].SummaryNew.Value.HumanAttributes.HasHat=2 items[0].SummaryNew.Value.HumanAttributes.HasBag=1 items[0].SummaryNew.Value.HumanAttributes.Sex=Man items[0].SummaryNew.Value.HumanAttributes.Age=30 items[0].SummaryNew.Value.HumanAttributes.HairStyle=2 items[0].SummaryNew.Value.HumanAttributes.HasUmbrella=1 items[0].SummaryNew.Value.HumanAttributes.Bag=0 items[0].SummaryNew.Value.HumanAttributes.Cap=2 items[0].SummaryNew.Value.FaceAttributes.Sex=Man items[0].SummaryNew.Value.FaceAttributes.Age=35 items[0].SummaryNew.Value.FaceAttributes.Mask=0 items[0].SummaryNew.Value.FaceAttributes.Beard=1 items[0].SummaryNew.Value.FaceAttributes.Glass=2 items[0].SummaryNew.Value.FaceAttributes.Emotion=Smile items[0].SummaryNew.Value.FacePath=/mnt/2010/8/11/dav/15:40:50.jpg items[0].SummaryNew.Value.FaceScenePath=/mnt/2010/8/11/dav/15:40:51.jpg ... </pre>
Response	

Step 4 Close the finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.10.5 Find Media Files".

4.10.9 Find media files with TrafficCar info

Step 1 Create a media files finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and TrafficCar conditions with the finder

Table 4-120

URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findFile		
Method	GET		
Description	<p>Start to find media files satisfied the common conditions and TrafficCar conditions with the finder.</p> <p>Note: some request params can use compare condition, its value is an array, first item is a string of compare condition type, and the following items are compare values.</p> <p>Compare condition type can be:</p> <ul style="list-style-type: none"> "==": means equal, followed with one param, if param type is string, then param value can contain some "*" to match any string. " ": means equal one of the params, followed with one or more params. "<>": means inside range, followed by two integer param, "><": means outside range, followed by two integer param, 		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
condition	object	R	The search conditions
+Channel	int	R	The video channel to search, video channel index start from 1, use -1 to search all video channel.
+StartTime	string	O	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	O	End time to search, ex: "2010-06-25 12:05:00"
+Types	array<string>	O	Search file types
+Flags	array<string>	O	Search flags. It can be: "Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout"
+Events	array<string>	O	Search event list Ignored, use Event under DB param.
+Dirs	array<string>	O	Search directory list, if omit, search all. Each dir path max string length is 259.
+DB	object	R	DB object
++TrafficCar	object	R	DB filter for TrafficCar
+++PlateNumber	<compare condition>	O	PlateNumber condition, use compare condition format, ex: ["==", "*888"]
+++Event	<compare condition>	O	Event condition, ex: [" ", "TrafficGate", "Alarm*"]
+++Speed	<compare condition>	O	Speed condition, ex: ["<>", 40, 80]
+++PlateType	<compare condition>	O	Plate type condition, use compare condition format, ex: ["==", "Armed"],
+++PlateColor	<compare condition>	O	Plate color condition, use compare condition format, ex: ["==", "Blue"],
+++VehicleColor	<compare condition>	O	Vehicle color condition, use compare condition format, ex: ["==", "White"],
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.250/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[0]=jpg&condition.Flags[0]=Event&condition.DB.TrafficCar.		

	PlateNumber[0]=%3d%3d&condition.DB.TrafficCar.PlateNumber[1]=%2a888&condition.D B.TrafficCar.Speed[0]=%3c%3e&condition.DB.TrafficCar.Speed[1]=40&condition.DB.Traffi cCar.Speed[2]=80&condition.DB.TrafficCar.VehicleColor[0]=%3d%3d&condition.DB.Traffic Car.VehicleColor[1]=White
Response	OK

Step 3 Get the media file information found by the finder

Table 4-121

URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findNextFile		
Method	GET		
Description	Get the media file information found by the finder		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
count	int	R	Record num to get, should less than 100.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
found	int	R	Record num that found.
items	array<object>	R	Result record items.
+Channel	int	O	The video channel index start from 0,
+StartTime	string	R	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	R	End time to search, ex: "2010-06-25 12:05:00"
+Type	string	O	File type, should be: "jpg"
+Events	array<string>	O	Events with this record
+FilePath	string	O	File path, string max length is 259
+CutLength	int	O	File length that cut between start time and end time
+Length	int	O	File length of the whole file
+Summary	object	R	DB record info
++TrafficCar	object	R	TrafficCar record info
+++PlateNumber	string	R	Car plate number
+++PlateType	string	O	Plate type,
+++PlateColor	string	O	Plate color, ex: "Yellow", "Blue", ... etc
+++VehicleColor	string	O	Vehicle color, ex: "Yellow", "Blue", ... etc
+++Country	string	O	Country info. max string length is 19
+++Speed	int	O	Vehicle speed, unit is km/hour
+++Event	string	O	The event info, ex: "TrafficJunction"
[Example]			
Request	GET http://192.168.1.250/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100		

Response	<pre> found=100 items[0].Channel=1 items[0].StartTime=2011-1-1 12:00:00 items[0].EndTime=2011-1-1 13:00:00 items[0].Type=jpg items[0].Events[0]=TrafficJunction items[0].FilePath=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg items[0].Length=790 items[0].Summary.TrafficCar.PlateNumber=A08888 items[0].Summary.TrafficCar.PlateType=Unknown items[0].Summary.TrafficCar.PlateColor=Blue items[0].Summary.TrafficCar.VehicleColor=White items[0].Summary.TrafficCar.Country=China items[0].Summary.TrafficCar.Speed=70 items[0].Summary.TrafficCar.Event=TrafficJunction ... </pre>
----------	--

Step 4 Close the finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.10.5 Find Media Files".

4.10.10 Find media files with IVS info

Step 1 Create a media files finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and IVS conditions with the finder

Table 4-122

URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findFile		
Method	GET		
Description	Start to find media files satisfied the common conditions and IVS conditions with the finder.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
condition	object	R	The search conditions
+Channel	int	R	The video channel to search, video channel index start from 1, use -1 to search all video channel.
+StartTime	string	O	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	O	End time to search, ex: "2010-06-25 12:05:00"
+Types	array<string>	O	Search file types
+Flags	array<string>	O	Search flags. It can be: "Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout"

+Events	array<string>	O	Search event list,
+Dirs	array<string>	O	Search directory list, if omit, search all. Each dir path max string length is 259.
+DB	object	R	DB object
++IVS	object	R	DB filter for IVS
+++Rule	string	O	IVS rule condition. It can be: "CrossLineDetection" , "CrossRegionDetection" , "LeftDetection" , "WanderDetection" , "MoveDetection" , "RioterDetection" , "CrossFenceDetection" , "TakenAwayDetection" , "PasteDetection" , "Preservation" , "StayDetection" , "TailDetection"
+++Action	string	O	IVS action. It can be: "Appear" , "Disappear" , "Inside" , "Cross"
+++ObjectType	array<string>	O	IVS object type, item in array can be: "Unknown" , "Human" , "Vehicle" , "NonMotor"
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.250/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[0]=jpg&condition.DB.IVS.Rule=CrossLineDetection&condition.DB.IVS.Action=Cross&condition.DB.IVS.ObjectType[0]=Human&condition.DB.IVS.ObjectType[1]=NonMotor		
Response	OK		

Step 3 Get the media file information found by the finder

Table 4-123

URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findNextFile		
Method	GET		
Description	Get the media file information found by the finder		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
count	int	R	Record num to get, should less than 100.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
found	int	R	Record num that found.
items	array<object>	R	Result record items.
+Channel	int	O	The video channel index start from 0,
+StartTime	string	R	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	R	End time to search, ex: "2010-06-25 12:05:00"
+Type	string	O	File type, should be: "jpg"
+Events	array<string>	O	Events with this record
+FilePath	string	O	File path, string max length is 259
+CutLength	int	O	File length that cut between start time and end time
+Length	int	O	File length of the whole file

+Summary	object	R	DB record info
++IVS	object	R	IVS record info
+++Rule	string	R	IVS rule, see above for valid value
+++Action	string	O	IVS action, see above for valid value
+++ObjectType	string	O	IVS object type, see above for valid value
[Example]			
Request	GET http://192.168.1.250/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100		
Response	found=100 items[0].Channel=1 items[0].StartTime=2011-1-1 12:00:00 items[0].EndTime=2011-1-1 13:00:00 items[0].Type=jpg items[0].Events[0]=CrossLineDetection items[0].FilePath=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg items[0].Length=790 items[0].Summary.IVS.Rule=CrossLineDetection items[0].Summary.IVS.Action=Cross items[0].Summary.IVS.ObjectType=Human ...		

Step 4 Close the finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.10.5 Find Media Files".

4.10.11 Find media files with NonMotor info

Step 1 Create a media files finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and NonMotor conditions with the finder

Table 4-124

URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findOne		
Method	GET		
Description	Start to find media files satisfied the common conditions and NonMotor conditions with the finder.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
condition	object	R	The search conditions
+Channel	int	R	The video channel to search, video channel index start from 1, use -1 to search all video channel.

+StartTime	string	O	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	O	End time to search, ex: "2010-06-25 12:05:00"
+Types	array<string>	O	Search file types
+Flags	array<string>	O	Search flags. It can be: "Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout"
+Events	array<string>	O	Search event list,
+Dirs	array<string>	O	Search directory list, if omit, search all. Each dir path max string length is 259.
+DB	object	R	DB object
++NonMotorRecordFilter	object	R	DB filter for NonMotor
+++NumOfCycling	int	O	Number of people that cycling, value can be 1 ~ 3
+++Color	string	O	NonMotor color, can be "White" "Orange" "Pink" "Black" "Red" "Yellow" "Gray" "Blue" "Green" "Purple" "Brown" "Sliver" "Darkviolet" "Maroon" "Dimgray" "Whitesmoke" "Darkorange" "Mistyrose" "Tomato" "Olive" "Gold" "Darkolivegreen" "Chartreuse" "Greencyellow" "Forestgreen" "Seagreen" "Deepskyblue" "Cyan" "Other"
+++Category	string	O	NonMotor type, can be : "Non-Motor" "Bicycle" "Tricycle" "Motorcycle" "DualTriWheelMotorcycle" "LightMotorcycle" "EmbassyMotorcycle" "MarginalMotorcycle" "AreaoutMotorcycle" "ForeignMotorcycle" "TrialMotorcycle" "CoachMotorcycle"
+++Helmet	int	O	Helmet status, 0 : unknown, 1 : without helmet, 2 : with helmet
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.250/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[0]=jpg&condition.DB.NonMotorRecordFilter.NumOfCycling=2&condition.DB.NonMotorRecordFilter.Color=White&condition.DB.NonMotorRecordFilter.Category=Bicycle&condition.DB.NonMotorRecordFilter.Helmet=1		
Response	OK		

Step 3 Get the media file information found by the finder

Table 4-125

URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findNextFile		
Method	GET		
Description	Get the media file information found by the finder		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
count	int	R	Record num to get, should less than 100.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description

found	int	R	Record num that found.
items	array<object>	R	Result record items.
+Channel	int	O	The video channel index start from 0,
+StartTime	string	R	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	R	End time to search, ex: "2010-06-25 12:05:00"
+Type	string	O	File type, should be: "jpg"
+Events	array<string>	O	Events with this record
+FilePath	string	O	File path, string max length is 259
+CutLength	int	O	File length that cut between start time and end time
+Length	int	O	File length of the whole file
+SummaryNew	object	R	NonMotor info
++Key	string	R	DB record name, should be " NonMotorRecordFilter "
++Value	object	R	DB record value
+++NumOfCycling	int	O	Number of people that cycling, value can be 1 ~ 3
+++Color	string	O	NonMotor color, can be "White" "Orange" "Pink" "Black" "Red" "Yellow" "Gray" "Blue" "Green" "Purple" "Brown" "Sliver" "Darkviolet" "Maroon" "Dimgray" "Whitesmoke" "Darkorange" "Mistyrose" "Tomato" "Olive" "Gold" "Darkolivegreen" "Chartreuse" "Greenyellow" "Forestgreen" "Seagreen" "Deepskyblue" "Cyan" "Other"
+++Category	string	O	NonMotor type, can be : "Non-Motor" "Bicycle" "Tricycle" "Motorcycle" "DualTriWheelMotorcycle" "LightMotorcycle" "EmbassyMotorcycle" "MarginalMotorcycle" "AreaoutMotorcycle" "ForeignMotorcycle" "TrialMotorcycle" "CoachMotorcycle"
+++Helmet	int	O	Helmet status, 0 : unknown, 1 : without helmet, 2 : with helmet
[Example]			
Request	GET http://192.168.1.250/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100		
Response	found=100 items[0].Channel=1 items[0].StartTime=2011-1-1 12:00:00 items[0].EndTime=2011-1-1 13:00:00 items[0].Type=jpg items[0].Events[0]=NonMotorDetect items[0].FilePath=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg items[0].Length=790 items[0].SummaryNew.Key=NonMotorRecordFilter items[0].SummaryNew.Value.NumOfCycling=2 items[0].SummaryNew.Value.Color=White items[0].SummaryNew.Value.Category=Bicycle items[0].SummaryNew.Value.Helmet=1 ...		

Step 4 Close the finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.10.5 Find Media Files".

4.10.12 Searching for Media Files According to WorkClothesDetection

Step 1 Create a media files finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and NonMotor conditions with the finder

URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findFile		
Method	GET		
Parameter Format	key=value format at URL		
Name	Type	R/O	Description
object	int32	R	Search object of the media files
condition	object	R	Search condition
+Channel	int32	R	Video channel number; integer; the video channel number starts from 1.
+StartTime	string	R	Start time of the search.
+EndTime	string	R	End time of the search.
+Types	array<string>	O	Type of search file. It should be jpg. ["jpg"]
+Flags	array<string>	O	Search condition, including Timing, Manual, Marker, Event, Mosaic, and Cutout. It should be Event.
+Events	array<string>	R	Search event list. It must be WorkClothesDetection. ["WorkClothesDetection"]
+Dirs	array<string>	O	Search directory list. If it does not exist, searches for all directories. ["/mnt/dvr/sda0", "/mnt/dvr/sda1"]
+DB	object	R	The filtering condition object of database.
++WorkClothesDetection	object	R	The filtering condition object of database for WorkClothesDetection.
+++Helmet	object	O	Safety helmet attribute. If this field does not exist, it means to search for attributes of all helmets.
++++HasHelmet	enumint8[6]	R	Wearing safety helmet. enumint8[] { 0: Unknown 1: No 2: Yes }
++++Helmet	array<string>	R	Safety helmet color. ["Red", "Orange"]

Color	ng>		For value range, see the following ColorEnum.	
++++IsCompliant	int32	O	Safety helmet alarm mode. 1: Mismatch attributes alarm; 2: Match attributes alarm.	1
+++Clothes	object	O	Work uniform attribute. If this field does not exist, it means to search for attributes of all work uniforms. It is optional.	
++++HasClothes	enumint8[]	R	Wearing work uniform. enumint8[]: 0: Unknown 1: No 2: Yes }	[1, 2]
++++ClothesColor	array<string>	R	Work uniform color. For value range, see the following ColorEnum.	["Red", "Orange"]
++++HasLegalClothes	int32	O	Work uniform alarm mode. 1: Mismatch attributes alarm; 2: Match attributes alarm.	1
++++TriggerMode	string	O	Detection mode. Attribute detection: TypeDetect; Registered database: CompareDetect.	"TypeDetect"
+++Mask	object	O	Face mask attribute. If this field does not exist, it means to search for attributes of all face masks.	
++++HasMask	int32	O	Wearing face mask. 1: No. 2: Yes.	1
+++Type	string	R	Wear type: Safety Helmet: Helmet. Work uniform: Clothes. Normal hat: NormalHat. Face mask: mask.	"Helmet"
+++NormalHat	object	O	Normal hat attribute. If this field does not exist, it means to search for attributes of all normal hats.	
++++HasNormalHat	enumint8[256]	O	enumint8[]: 0: Unknown 1: Not wearing normal hat 2: Wearing normal hat }	[1,2]
+++Glasses	object	O	Glasses attribute. If this field does not exist, it means to search for attributes of all glasses.	
++++HasGlasses	int32	O	Wearing glasses alarm mode. 0: Triggers an alarm when the target wears glasses. 1: Triggers an alarm	0

			when the target does not wear glasses.	
--	--	--	--	--

Request Example

GET

```
http://192.168.1.250/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[0]=jpg&condition.Flags[0]=Event&condition.Events[0]=WorkClothesDetection&condition.DB.WorkClothesDetection.Helmet.HasHelmet[0]=1&condition.DB.WorkClothesDetection.Helmet.HelmetColor[0]=Red&condition.DB.WorkClothesDetection.Type=Helmet
```

[Response Params] (OK)

Name	Type	R/O	Description	Example
Response Example				
OK				

Step 3 Get the media file information found by the finder

URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findNextFile			
Method	GET			
Parameter Format	key=value format at URL			
Name	Type	R/O	Description	Example
object	int32	R	Search object of the media files	08137
count	int32	R	The number of acquired results. It should be less than 100.	15

Request Example

```
GET http://192.168.1.250/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=15
```

[Response Params] (key=value format)

Name	Type	R/O	Description	Example
found	int32	R	The number of searched results.	100
items	object[]	R	Searched result array.	
+Channel	int32	O	Video channel number; integer; the video channel number starts from 0.	0
+StartTime	string	R	Start time of recording.	"2010-05-25 12:05:00"
+EndTime	string	R	End time of recording.	"2010-06-25 12:05:00"
+Type	string	O	File type. It should be jpg.	jpg
+Events	array<string>	O	Events related to the recordings.	["WorkClothesDetection"]
+FilePath	string	O	File path. It cannot exceed 259 characters.	"/mnt/2010/8/11/dav/15:40:50.jpg"
+CutLength	int32	O	The file length within a specified period.	79000
+Length	int32	O	The whole file length.	5536
+SummaryNew	object[]	R	Database record object.	

++Key	string	R	Database record name. It should be WorkClothesDetection.	"WorkClothesDetection"
++Value	object	O	Database record value.	
+++Helmet	object	O	Safety helmet attribute. Required.	
++++HasHelmet	enumint8	R	Wearing safety helmet enumint8{ 0: Unknown 1: No 2: Yes }	2
++++HelmetColor	string	R	Safety helmet color. For value range, see the following ColorEnum.	Red
+++Clothes	object	O	Safety helmet attribute.	
++++HasClothes	enumint8	R	Wearing work uniform. enumint8{ 0: Unknown 1: No 2: Yes }	2
++++ClothesColor	string	R	Work uniform color. For value range, see the following ColorEnum.	"Red"
+++HumanPath	string	R	Person thumbnail path.	"/picid/1.jpg"
+++HumanImageLength	uint32	R	Image size. Unit: Byte	123
+++HumanScenelImage	string	R	Panoramic image path corresponding to the person thumbnail.	"/picid/1.jpg"
+++HumanScenelImageLength	uint32	R	Image size. Unit: Byte.	123
+++HumanBoundingBox	Rect	R	Person enclosure box. Rectangular information.	[24, 16, 8152, 8144]
+++Glasses	object	O	Glasses attribute.	
++++GlassessType	enumint	R	Alarm type for glasses detection. enumint{ 0: No glasses 1. Sunglasses 2: Black-rimmed glasses 3: Half-frame glasses 4: Rimless glasses 5: Ordinary glasses 6: Industrial goggles }	1
++++HasGla	int32	R	Glasses detection results.	0

sses			0: Compliant 1: Non-compliance 2: Unknown.	
------	--	--	--	--

Response Example

```

found=100
items[0].Channel=1
items[0].StartTime=2011-1-1 12:00:00
items[0].EndTime=2011-1-1 13:00:00
items[0].Type=jpg
items[0].Events[0]=WorkClothesDetection
items[0].FilePath=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg
items[0].CutLength=79000
items[0].Length=5536
items[0].SummaryNew[0].Key=WorkClothesDetection
items[0].SummaryNew[0].Value.Helmet.HasHelmet=2
items[0].SummaryNew[0].Value.Helmet.HelmetColor=Red
items[0].SummaryNew[0].Value.HumanPath=/picid/1.jpg
items[0].SummaryNew[0].Value.HumanImageLength=123
items[0].SummaryNew[0].Value.HumanSceneImage=/picid/1.jpg
items[0].SummaryNew[0].Value.HumanSceneImageLength=123
items[0].SummaryNew[0].Value.HumanBoundingBox[0]=24
items[0].SummaryNew[0].Value.HumanBoundingBox[1]=16
items[0].SummaryNew[0].Value.HumanBoundingBox[2]=8152
items[0].SummaryNew[0].Value.HumanBoundingBox[3]=8144

```

Step 4 Close the finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.10.5 Find Media Files".

Appendix:

Name	ColorEnum	
Type	char[16]	
Parameter	Description	Remarks
"Other"	Unrecognized (unknown color)	(254,254,254) The value of 254 for R or G is dedicated to special colors, so be careful to use it separately.
"Unknown"	Unknown color (no detection, or it is not in the known color range).	(254,254,252)
"Black"	Black	0,0,0
"Silver"	Silver	192,192,192
"Gray"	Gray	128,128,128
"White"	White	255,255,255
"Maroon"	Chestnut	128,0,0

"Red"	Red	255,0,0
"Purple"	Purple	128,0,128
"Fuchsia"	Magenta (not supported)	255,0,255
"Green"	Green (the RGB value used by the algorithm for Green is 0, 255, 0).	0,128,0
"Lime"	Green (the algorithm names RGB value (0, 255, 0) as Green, and the standard name is Lime, so Lime can be treated as green in actual use).	0,255,0
"Olive"	Olive	128,128,0
"Yellow"	Yellow	255,255,0
"Navy"	Navy (not supported)	0,0,128
"Blue"	Blue	0,0,255
"Teal"	Navy (not supported)	0,128,128
"Aqua"	Aqua (not supported)	0,255,255
"Orange"	Orange (not supported)	255,165,0
"Pink"	Pink	255,192,203
"Brown"	Brown	165,42,42
"Darkviolet"	Dark violet	148,0,211
"Dimgray"	Dark gray	105,105,105
"Whitesmoke"	White smoke	245,245,245
"Darkorange"	Dark orange	255,140,0
"Mistyrose"	Light rosy	255,228,225
"Tomato"	Tomato red	255,99,71
"Gold"	Gold	255,215,0
"Darkolivegreen"	Dark olive green	85,107,47
"Chartreuse"	Yellow green	127,255,0
"Greenyellow"	Green yellow	173,255,47
"Forestgreen"	Forest green	34,139,34
"Seagreen"	Ocean green	46,139,87
"Deepskyblue"	Deep sky blue	0,191,255
"Cyan"	Cyan	0,255,255
"Lightgreen"	Light Green	144,238,144
"Orangered"	Orange red (not supported)	255,69,0

4.10.13 Download Media File with the File Name

Download a file by filename. The <**Filename**> in URL is got by chapter file finding.

Request URL	http://<server>/cgi-bin/RPC_Loadfile/< Filename >
Method	GET
Request Params (none)	

Request Example

```
http://192.168.1.250/cgi-bin/RPC_Loadfile/mnt/sd/2015-01-08/001/dav/19/19.57.12-19.58.25[M][0@0][0].dav
```

Response Params (binary data in body)

<binary data>: Binary data pack

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: Application/octet-stream

Content-Length: xxxx

<file data>

4.10.14 Download Media File between Times

Download the media data between start time and end time.

Request URL	http://<server>/cgi-bin/loadfile.cgi?action=startLoad			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	Channel number; starting from 1	1
startTime	string	R	Start time of the recording in the format of: yyyy-mm-dd hh:mm:ss	"2010-05-25 12:05:00"
endTime	string	R	End time of the recording in the format of: yyyy-mm-dd hh:mm:ss	"2010-05-25 12:10:00"
subtype	int	O	Stream type. If the parameter does not exist, the default value is 0. 0: Main stream 1: Sub stream 1 2: Sub stream 2	0
Types	string	O	The type of files that are being searched for	"dav"
Request Example				
http://192.168.1.250/cgi-bin/loadfile.cgi?action=startLoad&channel=1&startTime=2012-10-8%2013:00:01&endTime=2012-10-8%2014:00:01&subtype=0&Types=dav				

Response Params (binary data in body)

<binary data>: Binary data pack

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: Application/octet-stream

Content-Length: xxxx

<file data>

4.10.15 Encrypted Download Media File with the File Name

Table 4-126

Syntax	http://<server>/cgi-bin/RecordStreamInterleaved.cgi?action=attachStream&path=< File name >[&password=< password >]
Method	GET
Description	Encrypted Download a file by filename. To get filename by chapter file finding.
Example	http://172.29.2.241/cgi-bin/RecordStreamInterleaved.cgi?action=attachStream&path=/mnt/sd/2019-07-01/001/dav/12/12.36.16-12.36.26[F][0@0][0].dav
Success Return	HTTP Code: 200 OK Content-Type: Application/octet-stream Content-Length: <fileLength> Body: <data> <data>
Comment	Parameters in URL: Filename: name of media files which would be downloaded. password: password for encrypting media file data. if not set password, then use the preset password

4.10.16 Query Total Number of Alarms

Request URL	http://<server>/cgi-bin/api/mediaFileFind/getCount		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
Request Example			
{}			

Response Params (JSON format in body)			
Name	Type	R/O	Description
count	uint32	R	total number
Response Example			
{ "count": 100 }			

4.11 Log

4.11.1 Find Logs

- Whether or not found logs satisfied the conditions

Table 4-127

Syntax	http://<server>/cgi-bin/log.cgi?action=startFind&condition.StartTime=< start >&condition.EndTime=< end >[&condition.StartTimeRealUTC=< startRealUTC >&condition.EndTimeRealUTC=< endRealUTC >][&condition.Type=< type >]
Method	GET
Description	Start to find log.
Example	Find log between 2011-1-1 12:00:00 and 2011-1-10 12:00:00, UTC time between 2011-01-01T04:00:00Z and 2011-01-10T04:00:00Z. URL is: http://192.168.1.250/cgi-bin/log.cgi?action=startFind&condition.StartTime=2011-1-1%2012:00:00&condition.EndTime=2011-1-10%2012:00:00&condition.StartTimeRealUTC=2011-01-01T04:00:00Z&condition.EndTimeRealUTC=2011-01-10T04:00:00Z
Success Return	token=1 count=100
Comment	Parameters in URL: start/end : the start/end time of log. Format is: yyyy-mm-dd hh:mm:ss. startRealUTC/endRealUTC : Indicates the UTC time of log start/end, in the format: yyyy-mm-ddThh:mm:ssZ. And (start , end) are mutually exclusive; If both exist, use RealUTC time first. In response, there is a token for further log finding process. If token is greater than 0, logs are found; otherwise no logs are found. Type : log type. The range is { "System", "Config", "Event", "Storage", "Account", "Data", "File", "CourseRecord" }. count : The logcount for the search condition.

- Get the particular number of logs

Table 4-128

Syntax	http://<server>/cgi-bin/log.cgi?action=doFind&token=< TokenValue >&count=< logCount >
Method	GET
Description	Find log with token TokenValue and count logCount.
Example	http://192.168.1.250/cgi-bin/log.cgi?action=doFind&token=1&count=100
Success Return	found =2 items[0]. ReINDo =789 items[0]. Time =2011-05-20 11:59:10 items[0]. Type =ClearLog items[0]. User =admin items[1]. Detail .Compression=H.264->MJPG items[1]. Detail .Data=Encode items[1]. ReINDo =790

	items[1]. Time =2011-05-20 11:59:21 items[1]. TimeRealUTC =2011-05-20T03:59:21Z items[1]. Type =SaveConfig items[1]. User =System ...
Comment	Parameters in URL: The TokenValue is got by startFind in the above section, and logCount is the count of logs for this query. The maximum value of logCount is 100.

Appendix:

Field in Response	Description
found	Count of found log, found is 0 if no log is found.
User	User name.
Type	Log type.
Time	Time of this log.
TimeRealUTC	日志 UTC 时间, TZ 格式
RelINDo	Log number.
Detail	Log details.

3. Stop query logs

Table 4-129

Syntax	http://<server>/cgi-bin/log.cgi?action=stopFind&token=< TokenValue >
Method	GET
Description	Stop query log by token TokenValue .
Example	http://192.168.1.250/cgi-bin/log.cgi?action=stopFind&token=1
Success Return	OK
Comment	Parameters in URL: The TokenValue is got by startFind in above section

4.11.2 Clear All the Logs

Table 4-130

Syntax	http://<server>/cgi-bin/log.cgi?action=clear
Method	GET
Description	Clear all the logs.
Example	http://192.168.1.250/cgi-bin/log.cgi?action=clear
Success Return	OK
Comment	—

4.11.3 Backup Logs

Table 4-131

Syntax	http://<server>/cgi-bin/Log.backup?action=All&condition.StartTime=< startTime >&condition.EndTime=< endTime >
Method	GET
Description	Download the log information between the start time and the end time as a file named Log.Backup default.
Example	http://192.168.1.250/cgi-bin/Log.backup?action=All&condition.StartTime=2014-8-25%2000:02:32&condition.EndTime=2020-8-25%2001:02:32
Success Return	HTTP/1.1 200 OK CONTENT-LENGTH: 743087 CONNECTION: close Content-type: application/binarytet-stream; charset=utf-8 &w_User: default &Time: 2014-09-01 15:20:45 &Type: VideoLoss &Content: EventType: VideoLoss channel: <8> StartTime: 2014-09-01 15:20:45 ...
Comment	Parameters in URL: startTime/endTime : the start/end time when log info built. 24 hour Format, as: yyyy-mm-dd hh:mm:ss. For example: 2014-8-25 00:02:32 2020-8-25 01:02:32

4.11.4 Seek Find Logs

Syntax	http://<server>/cgi-bin/Log.cgi?action=doSeekFind&token=< TokenValue >&offset=< offsetValue >&count=< logCount >
Method	GET
Description	Find log with token TokenValue, offset offsetValue and count logCount.
Example	http://192.168.1.250/cgi-bin/Log.cgi?action=doSeekFind&token=46878&offset=200&count=200
Success Return	found =2 items[0]. RelNDo =789 items[0]. Time =2011-05-20 11:59:10 items[0]. Type =ClearLog items[0]. User =admin items[0]. Detail .Compression=H.264->MJPG

	<pre>items[0].Detail.Data=Encode items[1].ReIndo=790 items[1].Time=2011-05-20 11:59:21 items[1].Type=SaveConfig items[1].User=System items[1].Detail.Compression=H.264->MJPG items[1].Detail.Data=Encode ... </pre>
Comment	<p>Parameters in URL:</p> <p>The TokenValue is got by startFind in the above section, and logCount is the count of logs for this query. offsetValue is the number which count from the first matched. The maximum value of logCount is 100.</p>

4.11.5 Export Encryped Log

Syntax	<code>http://<server>/cgi-bin/Log.exportEncrypedLog?action=All&condition.StartTime=<startTime>&condition.EndTime=<endTime>&condition.Password=<password></code>
Method	GET
Description	Get encrypted log in zip format
Example	<code>http://192.168.1.250/cgi-bin/Log.exportEncrypedLog?action=All&condition.StartTime=2014-8-25%2000:02:32&condition.EndTime=2020-8-25%2001:02:32&condition.Password=12345</code>
Success Return	<p>HTTP/1.1 200 OK Transfer-Encoding: chunked Connection: keep_alive Content-type: application/binarytet-stream; charset=utf-8 The binary data of encrypted log in zip format</p>
Comment	<p>StartTime/EndTime: the start/end time when log info built. 24 hour Format, as: yyyy-mm-dd hh:mm:ss. For example: 2014-8-25 00:02:32 2020-8-25 01:02:32 The Password is the input parameter for encrypted log in zip format. The binary data should be saved as .zip format</p>

4.11.6 [Config] Serial Port Log Redirection

Config Data Params				
Name	Type	R/O	Description	Example
DebugInfoRedir	object	O	The configuration of serial port log redirection.	
+SerialPortLogLevel	uint8	否	logs which need to be redirected. (default is 4)	4

			0 : Fatal error; 1: Normal ; 2 : Warning;3 : Info, 4 : The invoke path and data flow tracking in debug level 1 ; 5 : The key logic tracking in debug level 2; 6 : Other detail information in debug level 3; 7 :Debug info for legacy compatible)	
+SerialPortLogType	uint8	否	log type which need to be redirected. (0 : main chip logs ; 1 : main chip + sub-chip logs)	0

Please refer to "4.2.1 Get and Set Configure" for more info about the get/set operation

Get Config Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&name=DebugInfoRedir
```

Get Config Response Example

```
table.DebugInfoRedir.SerialPortLogLevel=3
table.DebugInfoRedir.SerialPortLogType=1
```

Set Config Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&DebugInfoRedir.SerialPortLogLevel=
6&DebugInfoRedir.SerialPortLogType=1
```

Set Config Response Example

```
OK
```

4.12 Upgrader

4.12.1 Strat to Upgrade

Table 4-132

Syntax	http://<server>/cgi-bin/upgrader.cgi?action=uploadFirmware
Method	POST
Description	Use this message to upload the firmware, and when the device receiving all the data successfully, it will start to upgrade the device, and then use the getState method to get the state.

Example	<pre>POST /cgi-bin/upgrader.cgi?action=uploadFirmware HTTP/1.1 Host: 192.168.1.250 Connection: keep-alive Content-Type: multipart/form-data; boundary=-----8655433224198 Content-Length: xxxxxxxxx -----8655433224198 Content-Disposition:form-data;name="upgrade"; filename= "xxxxxx.bin" Content-Type: application/octet-stream Firmware data....</pre> <p>-----8655433224198-</p>
Success Return	OK
Comment	—

4.12.2 Get Upgrade State

Table 4-133

Syntax	http://<server>/cgi-bin/upgrader.cgi?action=getState
Method	GET
Description	Use this message to upload the firmware, and when the device receiving all the data successfully, it will start to upgrade the device, and then use the getState method to get the state.
Example	http://192.168.1.250/cgi-bin/upgrader.cgi?action=getState
Success Return	state. State =Upgrading state. Progress =45
Comment	<p>Parameters in Response:</p> <p>State: the state of the upgrade, it can be Preparing, Downloading, DownloadFailed, Upgrading, Invalid, Failed, Succeeded, Cancelled, and NotEnoughMemory.</p> <p>Progress: the progress of the upgrade.</p>

4.12.3 Set upgrader url

url with Upgrade packet address

Syntax	http://<server>/cgi-bin/upgrader.cgi?action=updateFirmwareByUrl[&Url=<Url>][&checkType=<checkType>][&checkSum=<checkSum>]			
Method	GET			
Parameter Format	key=value format at URL			
Parameter	Type	R/O	Description	Example
Url	string	R	Upgrade package address, also support ftp address	https://aaa/bbb/ccc/license.bin

checkType	int	R	check method enumint8{ 0: MD5 }	0
checkSum	string	O	Upgrade package checksum	f38ad920
Example				
http://<server>/cgi-bin/upgrader.cgi?action=updateFirmwareByUrl=http://aaa/bbb/ccc/license.bin&checkType=0&checkSum=f38ad920				

Parameter Format	OK in body			
Parameter	Type	R/O	Description	Example
Example				
OK				

4.12.4 Cancel Upgrade

Syntax	http://<server>/cgi-bin/upgrader.cgi?action=cancel			
Method	GET			
参数格式	key=value format at URL			
Parameter	Type	R/O	Description	Example
Example				
http://<server>/cgi-bin/upgrader.cgi?action=cancel				

Parameter Format	OK in body			
Parameter	Type	R/O	Description	Example
Example				
OK				

4.12.5 Checking Cloud Update Version

Check cloud update version

Request URL	http://<server>/cgi-bin/api/CloudUpgrader/check							
Method	POST							
Request Params (JSON format in body)								
Name	Type	R/O	Description	Example				
way	int32	R	Checking methods. enumint8{ 0: Updating by updating server through direct connection. 1: Checking through proxy server. 2: Detecting by acquiring cached test results. }	0				
proxy	object	O	Proxy server address, valid when way==1 (optional).					

+IP	char[40]	R	Network address	"10.1.2.3"
+Port	int32	O	Port	8080
Request Example				
{ "way" : 0, "proxy" : { "IP" : "10.1.2.3", "Port" : 8080 } }				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
info	object	O	Test results	
Status	enumchar[12]]	R	Update status. enumchar[12]{ "None": No update is detected "Regular": Regular update (It requires user confirmation, and can only be updated to a later version). "Emergency": Mandatory update (The device automatically performs detection and update, and it can also degrade to an earlier version). "Automatic": Automatic update (The device automatically updates when a new version is available. It is currently for custom use and needs to be enabled) }	"None"
+PackageType	enumchar[32]]	O	Update package type of new version. When State is not None, return enumchar[32]{ "all": All package. "ptz": PTZ main control package. "web" "logo" "custom" "gui" "pd" "data" "ptz_power": PTZ power. "ptz_light": PTZ light. "ptz_heater": PTZ heater. }	"all"
+OldVersion	char[64]	O	Old version, which needs to be returned when State is not None.	"0000"

+NewVersion	char[64]	O	New version, which needs to be returned when State is not None.	"0004"
+Attention	char[2048]	O	Updated content of the new update package.	"What is new"
+PackageUrl	char[256]	O	Download address of update package (required for agent upgrade).	"https://example.com/1.zip"
+Packageld	char[64]	O	Update package ID	"1d2ee7"
+CheckSum	char[64]	O	SHA-256 checksum of the update package	"F3D288AB"
+BuildTime	char[24]	O	Build time of update package	"08-10-2018 01:01:02"

Response Example

```
{
  "info": {
    "State": "None",
    "PackageType": "all",
    "OldVersion": "0000",
    "NewVersion": "0004",
    "Attention": "What is new",
    "PackageUrl": "https://example.com/1.zip",
    "Packageld": "1d2ee7",
    "CheckSum": "F3D288AB"
    "BuildTime": "08-10-2018 01:01:02"
  }
}
```

4.12.6 Performing Online Update

Perform online update

Request URL	http://<server>/cgi-bin/api/CloudUpgrader/execute			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
NewVersion	char[64]	O	If the cloud version is newer than the new version obtained in the last check, the update fails.	"2.42.00.001"
way	enumint8	R	Update method enumint8{ 0: Updating by updating server through direct connection. 1: Updating by proxy server }.	0
proxy	object	O	Proxy server address, which is valid when way=1	

			(optional).	
+IP	char[40]	O	Network address	"10.1.2.3"
+Port	uint16	O	Port	8080
info	object	O	Update package information, which is needed when way==1 (optional).	
+PackageUrl	char[256]	O	Download address of the update package (required for agent upgrade).	"https://example.com/1.zip"
+Packageld	char[64]	O	Update package ID	"1d2ee7"
+CheckSum	char[64]	O	SHA-256 checksum of the update package	"F3D288AB"

Request Example

```
{
    "NewVersion" : "2.42.00.001",
    "way" : 0,
    "proxy" : {
        "IP" : "10.1.2.3",
        "Port" : 8080
    },
    "info" : {
        "PackageUrl" : "https://example.com/1.zip",
        "Packageld" : "1d2ee7",
        "CheckSum" : "F3D288AB"
    }
}
```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				
{}				

4.12.7 Canceling Online Update

Cancel the online update during the download process. If you have already started writing Flash, you cannot cancel the update.

Request URL	http://<server>/cgi-bin/api/CloudUpgrader/cancel			
Method	GET			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Request Example				
{}				

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

4.13 Http Uploading

4.13.1 [Config] Active Image and Event Uploading

Configure the device to upload images and events to the specified address. Before configuration, the client should monitor in the specified address. Once the configuration is complete, the device would upload images and event data to this address using the command described in chapter "4.13.2 Active Image and Event Uploading".

The parameters for uploading image and event are as follows:

Config Data Params				
Name	Type	R/O	Description	Example
PictureHttpUpload	object	R	The parameters for uploading image and event	
+PictureHttpUpload	object	R	Enable or not	
++Enable	bool	R	Enable or not	true
++Type	char[16]	O	Authentication "basic": HTTP Basic Authentication "digest": HTTP Digest Authentication	"digest"
+UploadServerList	object[]	R	The list of servers that receives uploaded information	
++Address	char[128]	R	IP address or domain name of the server	192.168.1.208
++Port	int	O	Server port	80
++UserName	char[32]	O	Username	"abc"
++Password	char[128]	O	Password	"123"
++Uploadpath	char[128]	O	Upload path	"/example/handlepic.php"
++EventType	char[][32]	O	Code list for uploaded event	["CrossLineDetection", "FaceDetection"]

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=PictureHttpUpload
```

Get Config Response Example

```
table.PictureHttpUpload.PictureHttpUpload.Enable=true
table.PictureHttpUpload.PictureHttpUpload.Type=digest
table.PictureHttpUpload.UploadServerList[0].Address=192.168.1.208
table.PictureHttpUpload.UploadServerList[0].Port=80
table.PictureHttpUpload.UploadServerList[0].UserName=abc
table.PictureHttpUpload.UploadServerList[0].Password=123
```

```

table.PictureHttpUpload.UploadServerList[0].Uploadpath=/example/handlepic.php
table.PictureHttpUpload.UploadServerList[0].EventType[0]=CrossLineDetection
table.PictureHttpUpload.UploadServerList[0].EventType[1]=FaceDetection

```

Set Config Request Example

```

http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&PictureHttpUpload.PictureHttpUpload
.Enable=true&PictureHttpUpload.PictureHttpUpload.Type=digest&PictureHttpUpload.UploadServerList[
0].Address=192.168.1.208&PictureHttpUpload.UploadServerList[0].Port=80&PictureHttpUpload.Upload
ServerList[0].UserName=abc&PictureHttpUpload.UploadServerList[0].Password=123&PictureHttpUploa
d.UploadServerList[0].Uploadpath=/example/handlepic.php&PictureHttpUpload.UploadServerList[0].Ev
entType[0]=CrossLineDetection&PictureHttpUpload.UploadServerList[0].EventType[1]=FaceDetection

```

Set Config Response Example

OK

4.13.2 Active Image and Event Uploading

The device uploads images and event data to the specified address based on the client configuration. For details on the parameters for each event, please refer to the corresponding [Event] chapters. The IP address, port and URL of the uploaded target server is specified by the PictureHttpUpload. Each set of images and events are sent within one separate HTTP request, in which multiple of images and events are contained. It is uploaded through multipart.

Request URL	http://<Address>:<Port>/<Uploadpath_of_PictureHttpUpload>		
Method	POST		
Request Params (multipart ; JSON in body; binary data in body)			
Name	Type	R/O	Description
Channel	int	R	video channel, which start from 0
Time	char[32]	R	Snap picture time "2022-03-30 15:40:01"
Events	object[]	R	Event Information list
+Code	char[32]	R	Event Code "FaceRecognition"
+Action	char[16]	R	Event action, with the values of "Start", "Stop" "Pulse". "Pulse"
+Index	int	R	Event channel number, starting from 0. 0
+Data	object	R	For specific parameters of each event, please refer to the corresponding chapters for reference.

Request Example

POST http://192.168.1.208/example/handlepic.php HTTP/1.1

User-Agent: Device/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: closed

--<boundary>

Content-Type: application/json

Content-Length: <data length>

```
{
    "Channel" : 0,
    "Tlme" : "2022-03-30 15:40:01"
    "Events" : [ {
        "Code" : "FaceRecognition",
        "Action" : "Pulse",
        "Index" : 0
        "Data" : {
            "UTC" : 123456789,
            .....
        }
    }, ..., { } ]
}
--<boundary>
Content-Type: image/jpeg
Content-Length: <data length>

<jpeg data>
--<boundary>
...
--<boundary>--
```

Response Params (OK in body)

Response Example

OK

4.13.3 [Config] Active Event Uploading

Configure the device to upload event (without image) to the specified address. Before configuration, the client should monitor in the specified IP address. Once the configuration is complete, the device would upload event data to this address using the command described in chapter " 4.13.4 Active Event Uploading ".

The parameters for uploading event are as follows:

Config Data Params				
Name	Type	R/O	Description	Example
EventHttpUpload	object	R	Parameters for uploading event	
+EventHttpUpload	object	R	Whether to enable the function	
++Enable	bool	R	Whether to enable the function	true
++Type	char[16]	O	Authentication Type "basic":HTTP Basic Authentication "digest" :HTTP Digest Authentication	"digest"
+UploadServerList	object[]	R	The server list that receives uploaded information	

++Address	char[128]	R	IP address or domain name of the server	192.168.1.208
++Port	int	O	Server port	80
++UserName	char[32]	O	Username	"abc"
++Password	char[128]	O	Password	123"
++Uploadpath	char[128]	O	Upload path	"/example/handleevt.php"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=EventHttpUpload
```

Get Config Response Example

```
table.EventHttpUpload.EventHttpUpload.Enable=true
table.EventHttpUpload.EventHttpUpload.Type=digest
table.EventHttpUpload.UploadServerList[0].Address=192.168.1.208
table.EventHttpUpload.UploadServerList[0].Port=80
table.EventHttpUpload.UploadServerList[0].UserName=abc
table.EventHttpUpload.UploadServerList[0].Password=123
table.EventHttpUpload.UploadServerList[0].Uploadpath=/example/handleevt.php
```

Set Config Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&EventHttpUpload.EventHttpUpload.Enable=true&EventHttpUpload.EventHttpUpload.Type=digest&EventHttpUpload.UploadServerList[0].Address=192.168.1.208&EventHttpUpload.UploadServerList[0].Port=80&EventHttpUpload.UploadServerList[0].UserName=abc&EventHttpUpload.UploadServerList[0].Password=123&EventHttpUpload.UploadServerList[0].Uploadpath=/example/handleevt.php
```

Set Config Response Example

```
OK
```

4.13.4 Active Event Uploading

The device upload event data (without image) to the specified address based on the configuration of the client. For the specific parameter of each event, please refer to the corresponding [Event] chapters. The IP address, port and URL of the upload target server is specified by EventHttpUpload. Each event is sent within one separate HTTP request.

Request URL	http://<Address>:<Port>/<Uploadpath_of_EventHttpUpload>			
Method	POST			
Request Params (JSON in body)				
Name	Type	R/O	Description	Example
Code	char[32]	R	Event code	"FaceRecognition"
Action	char[16]	R	"Event action, with the values of "Start", "Stop""Pulse".	"Pulse"
Index	int	R	Event channel number, starting from 0	0
Data	object	R	For the specific parameters for each event, please refer to the corresponding chapters.	
Request Example				

Request Example
User-Agent: Device/1.0
Content-Type: application/json
Content-Length: <data length>

```
{
    "Code" : "FaceRecognition",
    "Action" : "Pulse",
    "Index" : 0
    Data
        "UTC" : 123456789,
        ...
    }
}
```

Response Params (OK in body)

Response Example

OK

4.13.5 [Config] Active Report Data Uploading

Configure the device to upload report data to the specified address. Before configuration, the client should monitor in the specified IP address. Once the configuration is complete, the device would upload report data to this address using the command described in chapters 4.13.6 ~ 4.13.11.

The parameters for uploading image are as follows:

Config Data Params				
Name	Type	R/O	Description	Example
ReportHttpUpload	object	R	parameters for uploading event	
+ReportHttpUpload	object	R	Whether to enable the function	
++Enable	bool	R	Whether to enable the function	true
++Type	char[16]	O	Authentication Type "basic" :HTTP basic authentication "digest" :HTTP digest authentication	"digest"
++Period	int	O	Upload period (unit: hour)	1
+UploadServerList	object[]	R	The server list that receives uploaded information	
++Address	char[128]	R	IP address or domain name of the server	192.168.1.208
++Port	int	O	Server port	80
++UserName	char[32]	O	Username	"abc"
++Password	char[128]	O	Password	123"
++Uploadpath	char[128]	O	Upload path	"/example/handleevt.php"
++ReportType	char [][40]	O	For the uploaded report data list, specific report data name,	["NumberStat", "ObjectDet ect"]

		please refer to the corresponding command reporting chapter description.	
--	--	--	--

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=EventHttpUpload
```

Get Config Response Example

```
table.ReportHttpUpload.ReportHttpUpload.Enable=true
table.ReportHttpUpload.ReportHttpUpload.Type=digest
table.ReportHttpUpload.ReportHttpUpload.Period=1
table.ReportHttpUpload.UploadServerList[0].Address=192.168.1.208
table.ReportHttpUpload.UploadServerList[0].Port=80
table.ReportHttpUpload.UploadServerList[0].UserName=abc
table.ReportHttpUpload.UploadServerList[0].Password=123
table.ReportHttpUpload.UploadServerList[0].Uploadpath=/example/handlerpt.php
table.ReportHttpUpload.UploadServerList[0].ReportType[0]=NumberStat
table.ReportHttpUpload.UploadServerList[0].ReportType[1]=ObjectDetect
```

Set Config Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&ReportHttpUpload.ReportHttpUpload.Enable=true&ReportHttpUpload.ReportHttpUpload.Type=digest&ReportHttpUpload.ReportHttpUpload.Period=1&ReportHttpUpload.UploadServerList[0].Address=192.168.1.208&ReportHttpUpload.UploadServerList[0].Port=80&ReportHttpUpload.UploadServerList[0].UserName=abc&ReportHttpUpload.UploadServerList[0].Password=123&ReportHttpUpload.UploadServerList[0].Uploadpath=/example/handlerpt.php&ReportHttpUpload.UploadServerList[0].ReportType[0]=NumberStat&ReportHttpUpload.UploadServerList[0].ReportType[1]=ObjectDetect
```

Set Config Response Example

OK

4.13.6 People Counting Report Data Uploading

If the client subscribes to the Numberstat (Number Statistics) report data through the ReportHttpUpload configuration, the device will use this command to upload the data to the specified address. The IP address, port and URL of the target server is specified by ReportHttpUpload. Each people counting report data is sent within one separate HTTP request.

Request URL	http://<Address>:<Port>/<Uploadpath_of_ReportHttpUpload>		
Method	POST		
Request Params (JSON in body)			
Name	Type	R/O	Description
NumberStat	Object[]	R	NumberStat: each dimension is a channel.
+SN	char[32]	R	Device serial number
+PeopleCount	Object[]	O	Humber statistics, the number of arrays is that of the cycles. If the period is an hour, then the number of array is one. If the

			period is three hours, then the number of array is three.	
++StartTime	char[20]	O	Start time	"2012-03-14 00:00:00"
EndTime:	char[20]	O	End time	"2012-03-14 23:59:59"
++EnteredSubtotal	uint	O	Subtotal of entered flow	56
++ExitedSubtotal	uint	O	Subtotal of exit flow	56
++AreaID	uint16 port	O	Area ID.	1
++Channel	int	O	Channel No.	0
+ManNumDetection	Object[]	O	Statistics of the number of the people in the region. The number of arrays equals to that of the period. If the time of period is an hour, then the number of array is one. If the time of the period is three hours, then the number of array is three.	
++StartTime	char[20]	O	Start time	"2012-03-14 00:00:00"
EndTime:	char[20]	O	End time	"2012-03-14 23:59:59"
++InsideSubtotal	uint	O	Used with ManNumDetection, QueueDetection, the subtotal of number of people in the region	56
++AreaID	uint16 port	O	Area ID.	1
++AverageStay Time	uint	O	Average regional retention time	10
++Channel	int	O	Channel No.	0
+QueueDetection	Object[]	O	Queuing Management Data. The number of arrays equals to that of the period. If the time of period is an hour, then the number of array is one. If the time of the period is three hours, then the number of array is three.	
++StartTime	char[20]	O	Start time	"2012-03-14 00:00:00"
++EndTime	char[20]	O	Statistics end time	"2012-03-14 23:59:59"
++InsideSubtotal	uint	O	Subtotal people in the area when ManNumDetection and QueueDetection are used	56
++AreaID	uint16	O	Area ID	1
++AverageStay Time	uint	O	Average stranding time in area	10

++Channel	int	O	Channel No.	0
Request Example				
POST http://192.168.1.208:80/example/handlerpt.php HTTP/1.1				
User-Agent: Device/1.0				
Content-Type: application/json				
Content-Length: <data length>				
<pre>{ "NumberStat" : [{ "SN" : "6M053FDYAQ00003" "ManNumDetection" : [{ "Channel" : 0, "EndTime" : "2021-03-11 17:59:59", "EnteredSubtotal" : 0, "ExitedSubtotal" : 0, "StartTime" : "2021-03-11 17:00:00" }, { "Channel" : 0, "EndTime" : "2021-03-11 6:59:59 PM", "EnteredSubtotal" : 0, "ExitedSubtotal" : 0, "StartTime" : "2021-03-11 6:00:00 PM" }, ..., { }], "PeopleCount" : [{ "Channel" : 0, "EndTime" : "2021-03-11 5:59:59 PM", "EnteredSubtotal" : 0, "ExitedSubtotal" : 0, "StartTime" : "2021-03-11 5:00:00 PM" }, { "Channel" : 0, "EndTime" : "2021-03-11 6:59:59 PM", "EnteredSubtotal" : 0, "ExitedSubtotal" : 0, "StartTime" : "2021-03-11 6:00:00 PM" }, ..., { }], "QueueDetection" : [{ "Channel" : 0, "EndTime" : "2021-03-11 5:59:59 PM", "EnteredSubtotal" : 0, "ExitedSubtotal" : 0, "StartTime" : "2021-03-11 5:00:00 PM" }, { "Channel" : 0, "EndTime" : "2021-03-11 6:59:59 PM", "EnteredSubtotal" : 0, "ExitedSubtotal" : 0, "StartTime" : "2021-03-11 6:00:00 PM" } } }</pre>				

```

        "StartTime" : "2021-03-11 6:00:00 PM"
    }, ..., { }]
}
}

```

Response Params (OK in body)

Response Example

OK

4.13.7 Video Structuring Report Data Upload

When the client subscribes ObjectDetect (video Structuring) dashboard, the device will upload video structuring dashboard to the specified address. The target service address and port, URL are designated by ReportHttpUpload. Each video structuring report data is sent within one separate HTTP request.

Request URL	http://<Address>:<Port>/<Uploadpath_of_ReportHttpUpload>			
Method	POST			
Request Params (JSON in body)				
Name	Type	R/O	Description	
ObjectDetect	Object[] []	R	Video structured data, the first dimension is channel, the second dimension is cycle count, the cycle is an hour, the array is 1, the cycle is 3 hours and the array is 3.	
SN	char[32]	R	Device SN	"1C03E08YAZ00020"
+UTC	uint	O	The end time of statistical Cycle (local UTC time)	1608508800
+UTCMS	uint	O	Milliseconds	0
+Period	int	O	Statistical cycle (unit: minute)	60
+PeriodBySeconds	int	O	Statistical cycle (unit: second)	60
+MotoVehicles	int	O	Traffic volume of motorized vehicle such as motorbike and tricycles	45
+Vehicles	int	O	Total count of motorized and non-motorized vehicle	0
+VehicleTypeFlow	object	O	Statistical data on the traffic volume based on vehicle type	
++PasserbyVehicles	int	O	Pedestrian traffic flow	0

Request Example

POST http://192.168.1.208:80/example/handlerpt.php HTTP/1.1

User-Agent: Device/1.0

Content-Type: application/json

Content-Length: <data length>

```
{
    "ObjectDetect" : [ [ {

```

```

    "MotoVehicles" : 8,
    "Period" : 60,
    "PeriodBySeconds" : 0,
    "SN" : "6J0CB81YAG10101",
    "UTC" : 1615492800,
    "UTCMS" : 0,
    "VehicleTypeFlow" : {
        "PasserbyVehicles" : 6
    },
    "Vehicles" : 17
}, ..., { }], ..., []]
}

```

Response Params (OK in body)

Response Example

OK

4.13.8 People Flow Heat Map Report Data Upload

When the client subscribes HeatMap (customer flow heat map report data, the device will upload video structuring data to the specified address. The target service address and port, URL are designated by ReportHttpUpload. Each people flow heat map report data is sent within one separate HTTP request.

Request URL	http://<Address>:<Port>/<Uploadpath_of_ReportHttpUpload>		
Method	POST		
Request Params (JSON in body)			
Name	Type	R/O	Description
HeatMap	Object[]	R	Heatmap is presented in binary format. Each dimension is a channel.
+Channel	int	O	Channel
+SN	char[32]	R	Device SN
+StartTime	char[20]	O	Statistics start Time
+EndTime	char[20]	O	Statistics end time
+width	uint	O	Image width
+height	uint	O	Image height
+EncodeData	string	O	base64 encoding of Heat map data

Request Example

POST http://192.168.1.208:80/example/handlerpt.php HTTP/1.1

User-Agent: Device/1.0

Content-Type: application/json

Content-Length: <data length>

```
{
    "HeatMap" : [ {

```

```

    "Channel" : 0,
    "SN" : "66:66:66:56:78:9a",
    "StartTime" : "2021-03-12 14:00:00",
    "EndTime" : "2021-03-12 15:00:00",
    "height" : 64,
    "width" : 64
    "EncodeData" : "abcd=",
    }, ..., {} ]
}

```

Response Params (OK in body)

Response Example

OK

4.13.9 ANPR Report Data Upload

When the client subscribes RoadFlowStat (road monitoring) report data, the device will upload video structuring data to the specified address. The target service address and port, URL are designated by ReportHttpUpload. Each ANPR report data is sent within one separate HTTP request.

Request URL	http://<Address>:<Port>/<Uploadpath_of_ReportHttpUpload>		
Method	POST		
Request Params (JSON in body)			
Name	Type	R/O	Description
RoadFlowStat	Object[] []	R	Road monitoring data, the first dimension is channel, the second dimension is cycle count. The cycle is an hour and the array is 1. The cycle is 3 hours and the array is 3.
SN	char[32]	O	Device SN
+UTC	uint	O	End time of statistical cycle
+UTCMS	uint	O	Milliseconds
+Lane	int	O	User defined lane number
+PresetID	int	O	PTZ preset, which must greater than 0.
+Period	int	O	Statistical cycle (unit: minute)
+PeriodBySec onds	int	O	Statistical cycle (unit: second)
+MotoVehicles	int	O	Traffic flow of non-motorized vehicles such as motorbike and tricycles.
+Vehicles	int	O	Total count of motorized and non-motorized vehicle
+VehicleTypeFlow	object	O	Statistical data on the traffic volume based on vehicle type
++PasserbyVe hicles	int	O	Pedestrian traffic flow

Request Example

POST http://192.168.1.208:80/example/handlerpt.php HTTP/1.1

User-Agent: Device/1.0

```

Content-Type: application/json
Content-Length: <data length>

{
    "RoadFlowStat" : [ [ {
        "SN" : "6J0CB81YAG10101",
        "UTC" : 1615485600,
        "UTCMS" : 0,
        "Lane" : 1,
        "PresetID" : 0,
        "Period" : 60,
        "PeriodBySeconds" : 0,
        "MotoVehicles" : 0,
        "Vehicles" : 26506
        "VehicleTypeFlow" : {
            "PasserbyVehicles" : 0
        },
    }, ... , { } ], ... , [ ]
}

```

Response Params (OK in body)

Response Example

OK

4.13.10 Crowd Distribution Report Data upload

When the client subscribes CrowdDistriMap (Crowd Distribution) report data, the device will upload video structuring data to the specified address. The target service address and port, URL are designated by ReportHttpUpload. Each crowd distribution report data is sent within one separate HTTP request.

Request URL	http://<Address>:<Port>/<Uploadpath_of_ReportHttpUpload>			
Method	POST			
Request Params (JSON in body)				
Name	Type	R/O	Description	Example
CrowdDistriMap	Object[]	R	Data statistics on crowd distribution. Each dimension is the data of a channel.	
SN	char[32]	O	Device SN	"1C03E08YAZ00020"
+CrowdDistriMap	object	O	Report Data on Crowd Distribution Map	
++DataList	object[64]	O	Return Crowd Distribution Map data list. The array size is 64 at most.	1
+++UTC	uint32	O	UTC time (local UTC time) when the data is recorded	6555478
+++AreaName	char[32]	O	Statistics Area name	"CMD-5"
+++PeopleNum	int	O	The number of people in statistics area	60

Request Example

POST http://192.168.1.208:80/example/handlerpt.php HTTP/1.1

User-Agent: Device/1.0

Content-Type: application/json

Content-Length: <data length>

```
{  
    "CrowdDistriMap" : [ {  
        "CrowdDistriMap" : {  
            "DataList" : [ {  
                "AreaName" : "CDM-1",  
                "PeopleNum" : 0,  
                "UTC" : 1617375600  
            }, {  
                "AreaName" : "CDM-1",  
                "PeopleNum" : 0,  
                "UTC" : 1617379200  
            } ]  
        },  
        "SN" : "7B0606BYAQ00010"  
    }, ... , {} ]  
}
```

Response Params (OK in body)

Response Example

OK

4.13.11 Vehicle Density Report Data Upload

When the client subscribes VehiclesDistri vehicle density) report data, the device will upload video structuring data to the specified address. The target service address and port, URL are designated by ReportHttpUpload. Each vehicle density report data is sent within one separate HTTP request.

Request URL	http://<Address>:<Port>/<Uploadpath_of_ReportHttpUpload>			
Method	POST			
Request Params (JSON in body)				
Name	Type	R/O	Description	Example
VehiclesDistri	Object[]	R	Data statistics on vehicle density distribution. Each dimension is the data of a channel.	
+SN	char[32]	O	Device SN	"1C03E08YAZ00020"
+CongestionDetection	object	O	The data report of vehicle density congestion rules.	0
++DataNum	uint32	O	Number of returned entires on vehicle congestion rules.	60
++DataList	object[]	O	The data list of vehicle congestion rules. The array size equal to DataNum	

			and the element is 64 at most.	
+++UTC	uint32	O	UTC time (local UTC time) when the data is recorded	3665789
+++RuleName	char[32]	O	Rule name	"VD-1"
+++VehiclesNum	int	O	The number of vehicles	32
+VehicleLimitDetection	object	O	The report data of traffic flow limit detection rules	0
++DataNum	uint32	O	Returned data entries on the traffic flow limit rules	1
++DataList	object[]	O	The data list of traffic flow limit detection rules. 64 elements at most.	
+++UTC	uint32	O	UTC time when the data is recorded	3665789
+++RuleName	char[32]	O	Rule name	
+++VehiclesNum	int	O	The number of vehicles	40

Request Example

POST http://192.168.1.208:80/example/handlerpt.php HTTP/1.1

User-Agent: Device/1.0

Content-Type: application/json

Content-Length: <data length>

```
{
  "VehiclesDistri" : [ {
    "CongestionDetection" : {
      "DataList" : [ {
        "RuleName" : "VD-1",
        "UTC" : 1617375600,
        "VehiclesNum" : 0
      }, {
        "RuleName" : "VD-1",
        "UTC" : 1617379200,
        "VehiclesNum" : 0
      } ],
      "DataNum" : 2
    },
    "SN" : "7B0606BYAQ00010",
    "VehicleLimitDetection" : {
      "DataNum" : 0
    }
  }, ... , {} ]
}
```

Response Params (OK in body)

Response Example

OK

4.13.12 FaceAnalysis Report Data Upload

When the client subscribes FaceAnalysis report data, the device will upload video structuring data to the specified address. The target service address and port, URL are designated by ReportHttpUpload. Each vehicle density report data is sent within one separate HTTP request.

Request URL	http://<Address>:<Port>/<Uploadpath_of_ReportHttpUpload>			
Method	POST			
Request Params (JSON in body)				
Name	Type	R/O	Description	Example
FaceAnalysis	Object[]	R	Data statistics on FaceAnalysis. Each dimension is the data of a channel.	
+SN	char[32]	R	Device SN	"CP0751AYAQ00004"
+Channel	uint	O	Channel	0
+Info	Object[]		Data statistics on FaceAnalysis. The number of arrays is the number of cycles. The cycle is one hour, the number of arrays is 1. The cycle is 3 hours, and the number of arrays is 3.	
++StartTime	systemtime	O	StartTime	"2022-12-21 11:00:00"
++EndTime	systemtime	O	EndTime	"2022-12-21 11:59:59"
++RepeatCount	uint32	O	Repeat Count	0
++FaceDetectCount	uint32	O	Statistics of face detection times after duplicate removal	102
++FaceAnalysisCount	uint32	O	Statistics of face analysis times after duplicate removal	92
Request Example				

POST http://192.168.1.208:80/example/handlerpt.php HTTP/1.1

User-Agent: Device/1.0

Content-Type: application/json

Content-Length: <data length>

```
{  
    "FaceAnalysis" : [  
        {  
            "Channel" : 0,  
            "SN" : "8J0751AYAQ00004",  
            "Info" : [{  
                "EndTime" : "2022-12-21 11:59:59",  
                "FaceAnalysisCount" : 92,  
                "FaceDetectCount" : 109,  
                "RepeatCount" : 0,  
                "StartTime" : "2022-12-21 11:00:00"  
            }, ...,{ }]  
    ]  
}
```

Response Params (OK in body)

Response Example

OK

5.Camera APIs.

5.1 Image

5.1.1 [Config] Brightness, Contrast and Saturation

- Get brightness, contrast and saturation

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=VideoColor
Method	GET
Description	Get brightness, contrast and saturation
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=VideoColor
Success Return	head.Name =Day, head.Brightness =50, head.Contrast =50, head.Saturation =50, head.Hue =50, head.Gamma =50, head.ChromaSuppress =50, head.Style =Standard, head.TimeSection =1 00:00:00-24:00:00
Comment	Parameters in URL: paramName and paramValue are as table below. In table below, head = table.VideoInSharpness [ChannelNo] [ConfigNo] ChannelNo : integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ConfigNo : array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

- Set brightness, contrast and saturation

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	SET
Description	Set brightness, contrast and saturation
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoColor[0][0].Brightness=50
Success Return	OK
Comment	Parameters in URL: paramName and paramValue are as table below. In table below, head = table.VideoInSharpness [ChannelNo] [ConfigNo]

	ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ConfigNo: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.
--	---

Appendix:

ParamName	ParamValue type	Description
head .Brightness	integer	Brightness, range is [0—100]
head .Contrast	integer	Contrast, range is [0—100]
head .Saturation	integer	Saturation, range is [0—100]

5.1.2 [Config] Sharpness

- Get sharpness

Table 5-1

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=VideoInSharpness
Method	GET
Description	Get sharpness
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=VideoInSharpness
Success Return	head.Level=4 head.Sharpness=8
Comment	Parameters in URL: paramName and paramValue are as table below. In table below, head =table.VideoInSharpness [ChannelNo] [ConfigNo] ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ConfigNo: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

- Set sharpness

Table 5-2

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	SET
Description	Set sharpness
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoInSharpness[0][0].Level=10&VideoInSharpness[0][0].Mode=1&VideoInSharpness[0][0].Sharpness=0
Success Return	OK
Comment	Parameters in URL: paramName and paramValue are as table below. In table below,

head =VideoInSharpness [ChannelNo] [ConfigNo] ChannelNo : integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ConfigNo : array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

Appendix:

ParamName	ParamValue type	Description
head .Sharpness	integer	Range is 0—100
head .Level	integer	Range is 0—100

5.1.3 [Config] Flip, Mirror and Rotate90

- Get flip, mirror and Rotate90

Table 5-3

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=VideoImageControl
Method	GET
Description	Get flip, mirror and Rotate90
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=VideoImageControl
Success Return	head.Flip=true head.Mirror=false head.Rotate90=0
Comment	Parameters in Response: head =table.VideoImageControl [ChannelNo] ChannelNo : integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ConfigNo : array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

- Set flip, mirror and Rotate90

Table 5-4

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	SET
Description	Set flip, mirror and Rotate90
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoImageControl[0].Flip=true
Success Return	OK
Comment	Parameters in URL: head =VideoImageControl [ChannelNo] ChannelNo : integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

	ConfigNo: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.
--	--

Appendix:

ParamName	ParamValue type	Description
head.Flip	bool	true: enable video flip function false: disable video flip function
head.Mirror	bool	true: enable video mirror function false: disable video mirror function
head.Rotate90	integer	Range is {0,1,2} Video rotation: 0: No rotate 1: clockwise rotate 90° 2: anticlockwise rotate 90°

5.2 Exposure

5.2.1 [Config] Exposure Config

- Get exposure

Table 5-5

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=VideoInExposure
Method	GET
Description	Get exposure
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=VideoInExposure
Success Return	head.AntiFlicker=0 head.Gain=50 head.GainMax=50 head.GainMin=0 head.Iris=50 head.IrisAuto=false head.Mode=0 head.Value1=40 head.Value2=40
Comment	Parameters in Response: head = table.VideoInExposure[ChannelNo][ConfigNo] ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ConfigNo: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

- Set exposure

Table 5-6

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	SET
Description	Set exposure
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoInExposure[0][0].Iris=50
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>head = VideoInExposure[ChannelNo][ConfigNo]</p> <p>ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p> <p>ConfigNo: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.</p>

Appendix:

ParamName	ParamValue type	Description
head .AntiFlicker	integer	Range is {0,1,2} AntiFlicker mode: 0: Outdoor 1: 50 Hz AntiFlicker 2: 60 Hz AntiFlicker
head .Gain	integer	Range is [0—100] If GainAuto is true, it's upper limit of auto gain, else it's the fixed gain adjust value.
head .GainMax	integer	Range is 0—100; the value must be greater than GainMin.
head .GainMin	integer	Range is 0—100; the value must be smaller than GainMax.
head .Iris	integer	Manual Iris setting. Range is 0—100.
head .IrisAuto	bool	Enable Iris automatically. true: IrisAuto false: No IrisAuto
head .Mode	integer	0: "Auto" by default 1: Low noise 2: Anti-smear 4: Manual (range) 5: Aperture priority 6: Manual (fixed) 7: Gain priority 8: Shutter priority 9: Flash light matching mode
head .Value1	float	Range is [0-1000], unit is millisecond If ExposureSpeed is 0(AutoExposure enable), it's

ParamName	ParamValue type	Description
		lower limit of AutoExposure time, otherwise it's time of manualExposure
head.Value2	float	Range is [0-1000], unit is millisecond Upper limit of AutoExposure time, should be bigger than ExposureValue1

5.3 Backlight

5.3.1 [Config] Backlight Config

- Get backlight

Table 5-7

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=VideoInBacklight
Method	GET
Description	Get backlight
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=VideoInBacklight
Success Return	head.GlareInhibition=50 head.Mode=Off head.WideDynamicRange=50
Comment	Parameters in Response: head = table.VideoInBacklight[ChannelNo][ConfigNo] ChannelNo : integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ConfigNo : array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

- Set backlight

Table 5-8

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	SET
Description	Set backlight
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoInBacklight[0][0].GlareInhibition=50
Success Return	OK
Comment	Parameters in URL: head = VideoInBacklight[ChannelNo][ConfigNo] ChannelNo : integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ConfigNo : array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

Appendix:

ParamName	ParamValue type	Description
<code>head.GlareInhibition</code>	integer	Range is 1–100.
<code>head.Mode</code>	char[32]	Off: Switched off Backlight: Backlight compensation GlareInhibition: HLC WideDynamic: WDR SSA: Scene adaptation
<code>head.WideDynamicRange</code>	integer	Range is 1–100.

5.4 White Balance

5.4.1 [Config] White Balance Config

- Get white balance

Table 5-9

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=VideoInWhiteBalance
Method	GET
Description	Get white balance
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=VideoInWhiteBalance
Success Return	head.GainBlue=50 head.GainRed=50 head.Mode=Auto
Comment	Parameters in Response: head = table.VideoInWhiteBalance[<i>ChannelNo</i>][<i>ColorConfigNo</i>] ChannelNo : integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ConfigNo : array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

- Set white balance

Table 5-10

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	SET
Description	Set white balance
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoInWhiteBalance[0][0].GainBlue=50
Success Return	OK
Comment	Parameters in URL:

	<p>head = VideoInWhiteBalance[ChannelNo][ConfigNo]</p> <p>ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p> <p>ConfigNo: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.</p>
--	--

Appendix:

ParamName	ParamValue type	Description
head .GainBlue	integer	Range is [0—100] Gain for blue value, Value is effective when WhiteBalance is "Custom."
head .GainRed	integer	Range is [0—100] Gain for red value, Value is effective when WhiteBalance is "Custom."
head .Mode	char[][][32]	Auto; Indoor; Outdoor; ATW; Manual; Sodium; Natural; StreetLamp; ManualDatum

5.5 Day-Night

5.5.1 [Config] Day-Night Config

- Get day-night

Table 5-11

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=VideoInDayNight
Method	GET
Description	Get day-night
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=VideoInDayNight
Success Return	head.Delay=10 head.Mode=Brightness head.Sensitivity=2 head.Type=Mechanism
Comment	<p>head = table.VideoInDayNight[ChannelNo][ConfigNo]:</p> <p>ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p> <p>ConfigNo: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.</p>

- Set Day-Night

Table 5-12

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue> [&<paramName>=<paramValue> ...]
Method	SET
Description	Set day-night

Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoInDayNight[0][0].Mode=BlackWhite
Success Return	OK
Comment	Parameters in URL: ParamName and paramValue are as table below. In table below, head = VideoInDayNight[<i>ChannelNo</i>][<i>ConfigNo</i>]

Appendix:

ParamName	ParamValue type	Description
head.Type	string	The range is {"Electron", "Mechanism", "NightICR", "Auto"}, the way of ICR switching.
head.Mode	string	The range is {"Color", "Brightness", "BlackWhite", "Photoresistor", "Gain"}. "Color": Always "color" "Brightness": Day/Night Auto "BlackWhite": Always black-and-white "Photoresistor": Switch according to photoresistor, "Gain": Switch according to gain
head.Sensitivity	integer	Range is [1-3]. Sensitivity of switching mode
head.Delay	integer	Range is [2-10]. Delay seconds when switching mode.

5.6 Zoom and Focus

To get the capability set of video input, refer to 4.5.12. For instance, you can use the following URL:

<http://192.168.1.250/cgi-bin/devVideoInput.cgi?action=getCaps&channel=1>

If ElectricFocus or SyncFocus of the return value is true, use 5.6.1 ~ 5.6.4; Otherwise, use 5.6.5 ~ 5.6.6 .

5.6.1 Adjust Focus

Adjust magnification and focus.

Request URL	http://<server>/cgi-bin/devVideoInput.cgi?action=adjustFocus			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	The video channel number which starts from 1, and the default value is 1.	1
focus	double	O	Relative stepping position of the vari-focal motor; range: [0~1]. -1 means resetting.	0.5
zoom	double	O	Relative stepping position of the zoom motor; range: [0~1]. -1 means resetting.	-0.5
Request Example				

```
http://192.168.1.250/cgi-bin/devVideoInput.cgi?action=adjustFocus&focus=0.5&zoom=-0.5
```

Response Params (OK in body)

Response Example

```
OK
```

5.6.2 Adjust Focus Continuously

Continuously adjust magnification and focus. Firstly, send a "non-zero" value to start zooming and adjusting the focal length to drive the motor to move, and then send "0" to stop the motor from moving. If only one of the operations is required, set the value of the other operation as -1.

Request URL	http://<server>/cgi-bin/devVideoInput.cgi?action=adjustFocusContinuously			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	The video channel number which starts from 1, and the default value is 1.	1
focus	double	O	Relative focusing rate; range: [-1,1]. -1: No operation 0: Stop Positive number: Moving forward Negative number: Moving backward	0.02
zoom	double	O	Relative zooming rate; range: [-1,1]. -1: No operation 0: Stop	-1
Request Example				
http://192.168.1.250/cgi-bin/devVideoInput.cgi?action=adjustFocusContinuously&channel=1&focus=0.02&zoom=-1				

Response Params (OK in body)

Response Example

```
OK
```

5.6.3 Auto Focus

Auto focus.

Request URL	http://<server>/cgi-bin/devVideoInput.cgi?action=autoFocus			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	The video channel number which starts from 1, and the default value is 1.	1
Request Example				
http://192.168.1.250/cgi-bin/devVideoInput.cgi?action=autoFocus&channel=1				

Response Params (OK in body)**Response Example**

OK

5.6.4 Get Focus Status

Get the focusing status.

Request URL	http://<server>/cgi-bin/devVideoInput.cgi?action=getFocusStatus		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	The video channel number which starts from 1, and the default value is 1.
Request Example			
http://192.168.1.250/cgi-bin/devVideoInput.cgi?action=getFocusStatus&channel=1			

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
status	object	R	Return the status information	
+Focus	double	R	Relative stepping position of the focusing motor; range: [0,1]	0.8
+Zoom	double	R	Relative stepping position of the zooming motor; range: [0,1]	0.3
+Status	char[16]	R	Focusing status "Normal": Normal "Autofocus": Auto focusing	"Normal"

Response Example

status.Focus=0.8

status.Zoom=0.3

status.Status=Normal

5.6.5 [Config] Zoom Config

Zooming configuration parameter:

Config Data Params				
Name	Type	R/O	Description	Example
VideolnZoom	object[][]	R	Zooming configuration parameter; two-dimensional array. The first dimension represents the video channel which starts from 1, and the second dimension represents the lighting scene. The beginning three elements constantly represent day, night and general scene.	
+Name	char[16]	O	Scene name	"Day"
+Speed	int	O	Zooming speed; range: [0–100]	8

+DigitalZoom	bool	O	Whether to enable digital zoom	true
--------------	------	---	--------------------------------	------

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=VideoInZoom
```

Get Config Response Example

```
table.VideoInZoom[0][0].Name=Day
table.VideoInZoom[0][0].Speed=8
table.VideoInZoom[0][0].DigitalZoom=true
...
```

Set Config Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoInZoom[0][0].DigitalZoom=false
&VideoInZoom[0][0].Speed=8
```

Set Config Response Example

```
OK
```

5.6.6 [Config] Focus Config

Vari-focal configuration parameter:

Config Data Params				
Name	Type	R/O	Description	Example
VideoInFocus	object[][]	R	Vari-focal configuration parameter; two-dimensional array. The first dimension represents the video channel which starts from 1, and the second dimension represents the lighting scene. The beginning three elements constantly represent day, night and general scene.	
+Name	char[16]	O	Scene name	"Day"
+Mode	int	O	Focus mode 2: Auto focus 3: Semi Auto (Customizable, uses auto focus first, and then locks the focusing module. Auto focus is not available at the time, and you have to manually adjust the focal length.) 4: Manual focus	2
+FocusLimit	int	O	Recommended limit value of near-field focusing; unit: mm. The value range depends on the device capability.	2000
+FocusFarLimit	int	O	Recommended limit value of far-field focusing; unit: mm.	5000

			The value range depends on the device capability.	
+AutoFocusTrace	int	O	Vari-focal tracking 0: Close 1: Open	1
+IRCorrection	int	O	IR light focusing and correction 0: No correction 1: Manual correction 2: Auto correction	1
+Sensitivity	int	O	Focusing sensitivity 0: High 1: Default 2: Low	1
+FocusLimitSelect Mode	char[16]	O	Focuings limit mode (The distance limit for near-field focusing) "Manual" "Auto"	"Auto"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=VideoInFocus
```

Get Config Response Example

```
table.VideoInFocus[0][0].Name=Day
table.VideoInFocus[0][0].Mode=0
table.VideoInFocus[0][0].Sensitivity=1
table.VideoInFocus[0][0].FocusLimitSelectMode=Manual,
table.VideoInFocus[0][0].FocusLimit=2000
table.VideoInFocus[0][0].FocusFarLimit=5000
table.VideoInFocus[0][0].AutoFocusTrace=0
table.VideoInFocus[0][0].IRCorrection=0
...
```

Set Config Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoInFocus[0][0].FocusLimit=2000
&VideoInFocus[0][0].FocusFarLimit=5000&VideoInFocus[0][0].Sensitivity=1
```

Set Config Response Example

```
OK
```

5.7 Lighting

5.7.1 [Config] Lighting Config

- Get lighting

Table 5-13

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=Lighting
Method	GET
Description	Get lighting
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=Lighting
Success Return	<pre>head.Correction=50 head.FarLight[0].Angle=0 head.FarLight[0].Light=0 head.Mode=ZoomPrio head.NearLight[0].Angle=0 head.NearLight[0].Light=0 head.MiddleLight[0].Angle=50 head.MiddleLight[0].Light=50</pre>
Comment	<p>head = table.Lighting[ChannelNo][ConfigNo] :</p> <p>ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p> <p>ConfigNo: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.</p>

- Set lighting

Table 5-14

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set lighting
Example	<p>Turn on light:</p> <p>http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&Lighting[0][0].FarLight[0].Light=10&Lighting[0][0].NearLight[0].Light=90&Lighting[0][0].Mode=Manual</p> <p>Shift the light to ZoomPrio mode:</p> <p>http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&Lighting[0][0].Correction=50&Lighting[0][0].Mode=ZoomPrio</p>
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>paramName and paramValue are as table below.</p> <p>In table below,</p> <p>head = Lighting[ChannelNo][ConfigNo]</p> <p>Some lighting equipment has the following restriction:</p> <p>Lighting[0][0].FarLight[0].Light + Lighting[0][0].NearLight[0].Light + Lighting[0][0].MiddleLight [0]. Light <=100.</p>

Appendix:

ParamName	ParamValue type	Description
-----------	-----------------	-------------

ParamName	ParamValue type	Description
head.Mode	string	Light mode. The range is {"Manual", "Auto", "Off", "ZoomPrio"}. The following are special for composite lamps. The range is { "Timing" , "SmartLight" , "ExclusiveManual" }
head.Correction	integer	Light compensation. The range is [0—100], effective in ZoomPrio mode.
head.FarLight[<i>Index</i>].Light	integer	Range is [0—100]. The luminance of far light.
head.MiddleLight[<i>Index</i>].Light	integer	Range is [0—100]. The luminance of middle light.
head.NearLight[<i>Index</i>].Light	integer	Range is [0—100]. The luminance of near light.

5.8 Video in Options

5.8.1 Change binocular camera's splice mode

Request URL	http://<server>/cgi-bin/api/MultiVideo/changeSpliceMode			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
State	int32	是	Splicing state 1: Switch to splice mode (Binocular camera's monocular mode) 0: Switch to non-splicing mode (Binocular camera's original mode)	0
Request Example				
{ "State": 0 }				
Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				{ }

5.8.2 [Config] Video in Options Config

It's not recommended to use the CGI command from "video in options"; It's now recommended to use the commands in 5.1 – 5.7.

- Get video in options

Table 5-15

Syntax	<code>http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=VideoInOptions</code>
Method	GET
Description	Get video in options config, such as Backlight, ExposureSpeed, DayNightColor, DayOptions, NightOptions, and NormalOptions.
Example	<code>http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=VideoInOptions</code>
Success Return	<pre> head.Backlight=0 head.DayNightColor=false head.ExposureSpeed=0 head.ExposureValue1=0.100000 head.ExposureValue2=80.000000 head.ExternalSync=0 head.ExternalSyncPhase=0 head.FlashControl.Mode=0 head.FlashControl.Pole=0 head.FlashControl.Value=0 head.FlashControl.PreValue=0 head.Flip=false head.Gain=50 head.GainAuto=true head.IrisAuto=false head.Mirror=false head.NightOptions.AntiFlicker=0 head.NightOptions.Backlight=0 head.NightOptions.BacklightRegion[0]=3096 head.NightOptions.BacklightRegion[1]=3096 head.NightOptions.BacklightRegion[2]=5096 head.NightOptions.BacklightRegion[3]=5096 head.NightOptions.BrightnessThreshold=50 head.NightOptions.DayNightColor=2 head.NightOptions.ExposureMode=0 head.NightOptions.ExposureSpeed=0 head.NightOptions.ExposureValue1=0 head.NightOptions.ExposureValue2=40 head.NightOptions.ExternalSyncPhase=125 head.NightOptions.Flip=false head.NightOptions.Gain=50 head.NightOptions.GainAuto=true head.NightOptions.GainBlue=50 </pre>

```
head.NightOptions.GainGreen=50
head.NightOptions.GainMax=50
head.NightOptions.GainMin=0
head.NightOptions.GainRed=50
head.NightOptions.GlareInhibition=0
head.NightOptions.IrisAuto=true
head.NightOptions.Mirror=false
head.NightOptions.Profile=3
head.NightOptions.ReferenceLevel=50
head.NightOptions.Rotate90=0
head.NightOptions.SunriseHour=0
head.NightOptions.SunriseMinute=0
head.NightOptions.SunriseSecond=0
head.NightOptions.SunsetHour=23
head.NightOptions.SunsetMinute=59
head.NightOptions.SunsetSecond=59
head.NightOptions.SwitchMode=4
head.NightOptions.WhiteBalance=Auto
head.NightOptions.WideDynamicRange=0
head.NightOptions.WideDynamicRangeMode=0
head.NormalOptions.AntiFlicker=0
head.NormalOptions.Backlight=0
head.NormalOptions.BacklightRegion[0]=3096
head.NormalOptions.BacklightRegion[1]=3096
head.NormalOptions.BacklightRegion[2]=5096
head.NormalOptions.BacklightRegion[3]=5096
head.NormalOptions.BrightnessThreshold=50
head.NormalOptions.DayNightColor=1
head.NormalOptions.ExposureMode=0
head.NormalOptions.ExposureSpeed=0
head.NormalOptions.ExposureValue1=0
head.NormalOptions.ExposureValue2=40
head.NormalOptions.ExternalSyncPhase=125
head.NormalOptions.Flip=false
head.NormalOptions.Gain=50
head.NormalOptions.GainAuto=true
head.NormalOptions.GainBlue=50
head.NormalOptions.GainGreen=50
head.NormalOptions.GainMax=50
head.NormalOptions.GainMin=0
head.NormalOptions.GainRed=50
head.NormalOptions.GlareInhibition=0
head.NormalOptions.IrisAuto=true
head.NormalOptions.Mirror=false
head.NormalOptions.Profile=0
head.NormalOptions.ReferenceLevel=50
head.NormalOptions.Rotate90=0
```

	<pre> head.NormalOptions.SunriseHour=0 head.NormalOptions.SunriseMinute=0 head.NormalOptions.SunriseSecond=0 head.NormalOptions.SunsetHour=23 head.NormalOptions.SunsetMinute=59 head.NormalOptions.SunsetSecond=59 head.NormalOptions.SwitchMode=0 head.ReferenceLevel=50 head.ReferenceLevelEnable=false head.Rotate90=0 head.SignalFormat=BT656 head.WhiteBalance=Disable </pre>
Comment	<p>Parameters in Response:</p> <p>head = table.VideoInOptions[ChannelNo]</p> <p>ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p>

- Set video in options

Table 5-16

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&< paramName >=< paramValue >[&< paramName >=< paramValue >...]
Method	GET
Description	Set video in options config, such as Backlight, ExposureSpeed, DayNightColor, DayOptions, NightOptions, and NormalOptions.
Example	<p>Set Auto Exposure:</p> <p>http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].ExposureMode=0&VideoInOptions[0].ExposureSpeed=0</p> <p>Set Low Noise:</p> <p>http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].ExposureMode=1&VideoInOptions[0].ExposureSpeed=0&VideoInOptions[0].GainMin=0&VideoInOptions[0].GainMax=60</p> <p>Set Low Motion Blur:</p> <p>http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].ExposureMode=2&VideoInOptions[0].ExposureSpeed=0&VideoInOptions[0].GainMin=0&VideoInOptions[0].GainMax=50&VideoInOptions[0].ExposureValue1=0&VideoInOptions[0].ExposureValue2=20</p> <p>Set Manual:</p> <p>http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].ExposureMode=4&VideoInOptions[0].ExposureSpeed=32&VideoInOptions[0].GainMin=0&VideoInOptions[0].GainMax=50&VideoInOptions[0].ExposureValue1=40&VideoInOptions[0].ExposureValue2=40</p> <p>Set SmartIRExposure:</p>

	<p>http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].SmartIRExposure=true</p> <p>Set Video Rotate:</p> <p>Flip:</p> <p>http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].Flip=true</p> <p>Mirror:</p> <p>http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].Mirror=true</p> <p>Or turn 90°:</p> <p>http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].Rotate90=1</p> <p>Set White Balance:</p> <p>http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].WhiteBalance=Night</p> <p>Or</p> <p>http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].WhiteBalance=Custom&VideoInOptions[0].GainRed=50&VideoInOptions[0].GainBlue=50&VideoInOptions[0].GainGreen=50</p> <p>(Sometimes you should set mode first before set GainRed or GainBlue: http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].WhiteBalance=Custom)</p>
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>In table below, head = VideoInOptions[ChannelNo]</p> <p>ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p>

Appendix:

ParamName	ParamValue type	Description
head.Backlight	integer	<p>Range is [0—n] n depends on capability in GetVideoInputCaps</p> <p>0: backlight closed. 1: backlight grade 1 ... n - backlight grade n</p>
head.DayNightColor	integer	<p>Range is {0,1,2}</p> <p>0: always multicolor 1: autoswitch along with brightness, 2: always monochrome</p>

ParamName	ParamValue type	Description
head.ExposureMode	integer	Range is {0,1,2, 4} 0: AutoExposure 1: Gain first 2: Exposure first 4: Manual.
head.ExposureSpeed	integer	Range is [0 — n+1] n depends on capability in GetVideoInputCaps 0: AutoExposure 1-n-1: manual Exposure grade n: AutoExposure with time limit. n+1: manualExposure with user-defined time (n is supported maximum exposure grade)
head.ExposureValue1	float	Range is [0.1—80], unit is millisecond If ExposureSpeed is 0(AutoExposure enable), it's lower limit of AutoExposure time, otherwise it's time of manualExposure
head.ExposureValue2	float	Range is [0.1—80], unit is millisecond Upper limit of AutoExposure time, should be bigger than ExposureValue1
head.ExternalSync	integer	Range is {0,1} External Synchronous 0: Internal Synchronization 1: External Synchronous
head.ExternalSyncPhase	float	Range is [0°—360°] External Synchronous Signal Phase
head.SmartIRExposure	bool	true: enable, false: disable
head.FlashControl.Mode	integer	Range is {0,1,2} 0: forbid flash 1: always flash 2: auto flash
head.FlashControl.Pole	integer	Range is {0,1, 2, 3} Trigger mode: 0: low level 1: high level 2: rising-edge 3: falling-edge
head.FlashControl.Value	integer	Range is [0—15] Flashlight time-unit: 0: 0us, 1: 64us, 2: 128us, 3: 192us ... 15 - 960us

ParamName	ParamValue type	Description
head.FlashControl.PreValue	integer	Range is [0—100] It is threshold of brightness value: if brightness is less than this value, flash light will begin to work.
head.Flip	bool	true: enable video flip function false: disable video flip function
head.Gain	integer	Range is [0—100] If GainAuto is true, it's upper limit of auto gain, else it's the fixed gain adjust value.
head.GainBlue	integer	Range is [0—100] Gain for blue value, Value is effective when WhiteBalance is "Custom."
head.GainRed	integer	Range is [0—100] Gain for red value, Value is effective when WhiteBalance is "Custom."
head.GainGreen	integer	Range is [0—100] Gain for green value, Value is effective when WhiteBalance is "Custom."
head.GainAuto	bool	true: GainAuto false: No GainAuto
head.IrisAuto	bool	true: IrisAuto false: No IrisAuto
head.Mirror	bool	true: enable video mirror function false: disable video mirror function
head.WhiteBalance	String	White balance Mode. Range is {Disable, Auto, Custom, Sunny, Cloudy, Home, Office, Night} Some IPC supports common modes: "Disable", "Auto", "Sunny", "Night", "Outdoor", "Custom" Sometimes the device support other advanced modes: "CustomColorTemperature", "Indoor", "ATW", "Manual", "AutoOutdoor", "ManualDatum" and so on.
head.ReferenceLevel	integer	Range is [0—100] The expected average brightness level of video frames.
head.Rotate90	integer	Range is {0,1,2} Video rotation: 0: No rotate 1: clockwise rotate 90° 2: anticlockwise rotate 90°
head.SignalFormat	String	Range is {Inside, BT656, 720p, 1080p, 1080i, 1080sF} Input Signal Mode

ParamName	ParamValue type	Description
head.AntiFlicker	integer	Range is {0,1,2} AntiFlicker mode: 0: Outdoor 1: 50 Hz AntiFlicker 2: 60 Hz AntiFlicker
head.GlareInhibition	integer	Range is [0—100] GlareInhibition: 0: Close GlareInhibition.
head.NightOptions.BrightnessThreshold	integer	NightOptions contain a set of parameters used when brightness is not enough. Range is [0—100] when brightness is less than the BrightnessThreshold, parameters change to Nightoptions.
head.NightOptions.IrisAuto	bool	true: IrisAuto false: No IrisAuto
head.NightOptions.SunriseHour	integer	Range is [00—23] Sunrise hour.
head.NightOptions.SunriseMinute	integer	Range is [00—59] Sunrise minute
head.NightOptions.SunriseSecond	integer	Range is [00—59] Sunrise second
head.NightOptions.SunsetHour	integer	Sunset time. Its range is same with sunrise time, and it should be after sunrise time.
head.NightOptions.SunsetMinute	integer	NightOptions are used if time is after sunset time and before sunrise time.
head.NightOptions.SunsetSecond	integer	
head.NightOptions.SwitchMode	integer	Range is {0,1,2} 0: NoSwitch, always use day options. 1: Switch depends on brightness. 2: Switch depends on time, switch to NightOptions when time is after sunset time and before sunrise. 3: NoSwitch, always use NightOptions. 4: No switch, always use NormalOptions.
head.NightOptions.Profile	integer	Range is {0,1,2,3} 0: use temporary day options. 1: use temporary NightOptions. 2: use temporary NormalOptions. 3: depends on <i>head.NightOptions.SwitchMode</i>
head.NightOptions.ExposureSpeed	integer	Range is the same as relevant items of day options in this table. Example:
head.NightOptions.ExposureValue1	float	Value range of <i>head.NightOptions.ExposureSpeed</i> is the same with <i>head.ExposureSpeed</i> .
head.NightOptions.ExposureValue2	float	

ParamName	ParamValue type	Description
head.NightOptions.Gain	integer	
head.NightOptions.GainAuto	bool	
head.NightOptions.GainBlue	integer	
head.NightOptions.GainGreen	integer	
head.NightOptions.GainRed	integer	
head.NightOptions.WhiteBalance	String	
head.NightOptions.ReferenceLevel	integer	
head.NightOptions.ExternalSyncPhase	float	
head.NightOptions.AntiFlicker	integer	
head.NightOptions.Backlight	integer	
head.NightOptions.DayNightColor	integer	
head.NightOptions.ExposureMode	integer	
head.NightOptions.GlareInhibition	integer	
head.NightOptions.Mirror	integer	
head.NightOptions.Flip	integer	
head.NightOptions.Rotate90	integer	
head.NormalOptions.BrightnessThreshold	integer	
head.NormalOptions.IrisAuto	bool	
head.NormalOptions.SunriseHour	integer	
head.NormalOptions.SunriseMinute	integer	
head.NormalOptions.SunriseSecond	integer	
head.NormalOptions.SunsetHour	integer	
head.NormalOptions.SunsetMinute	integer	
head.NormalOptions.SunsetSecond	integer	
head.NormalOptions.ExposureSpeed	integer	
head.NormalOptions.ExposureValue1	float	
head.NormalOptions.ExposureValue2	float	

NomalOptions contain a set of parameters similar with NightOptions.

Range is the same as relevant items of NightOptions in this table.

ParamName	ParamValue type	Description
head.NormalOptions.Gain	integer	
head.NormalOptions.GainAuto	bool	
head.NormalOptions.GainBlue	integer	
head.NormalOptions.GainGreen	integer	
head.NormalOptions.GainRed	integer	
head.NormalOptions.WhiteBalance	String	
head.NormalOptions.ReferenceLevel	integer	
head.NormalOptions.ExternalSyncPhase	float	
head.NormalOptions.AntiFlicker	integer	
head.NormalOptions.Backlight	integer	
head.NormalOptions.DayNightColor	integer	
head.NormalOptions.ExposureMode	integer	
head.NormalOptions.GlareInhibition	integer	
head.NormalOptions.Mirror	integer	
head.NormalOptions.Flip	integer	
head.NormalOptions.Rotate90	integer	

6.Storage API

6.1 Storage Devices

6.1.1 Get Hard Disk Information

Get the hard disk information.

Request URL	http://<server>/cgi-bin/storageDevice.cgi?action=factory.getPortInfo		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.250/cgi-bin/storageDevice.cgi?action=factory.getPortInfo			

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
into	object	R	Return the hard disk information	
+Total	uint	O	The number of ports of the bus, including the expansion bus.	2
+Plug	uint	O	The number of the mounted IDE (damaged IDEs are excluded)	1
+Mask	uint64	O	Indicates whether there is an IDE (including damaged IDE) mounted on the bus channel. Each bit represents the main IDE or the sub IDE of the bus channel. If the bit is set to "1", there is a mounted IDE. If the bit is set to "0", there is no mounted IDE. For example, bit 0 represents the main IDE of IDE0, bit1 represents the sub IDE of IDE0, bit2 represents the main IDE of IDE1, bit3 represents the sub IDE of IDE1, and more.	1
+Bad	uint	O	Indicates whether there is a damaged IDE mounted on each channel. Each bit represents the main or sub IDE of each channel. If the bit is set to "1", there is a damaged IDE. If the bit is set to "0", there is no damaged IDE. See Mask for the corresponding relationship.	0
+IDE	uint	O	The number of mounted IDE	1
+Esata	uint	O	The total port number of the eSATA	4
Response Example				
info.Total=2				
info.Plug=1				

```

info.Mask=1
info.Bad=0
info.IDE=1
info.Esata=4

```

6.1.2 Get the Name of All Storage Devices

Get the name of all storage devices.

Request URL	http://<server>/cgi-bin/storageDevice.cgi?action=factory.getCollect		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.250/cgi-bin/storageDevice.cgi?action=factory.getCollect			

Response Params (key=value format in body)			
Name	Type	R/O	Description
list	char[][64]	R	Return the name of the storage device
Response Example			
list[0]="/dev/sda0" list[1]="/dev/sda1" list[2]="/dev/sg1"			

6.1.3 Get Storage Device Information

Get all the storage device information .

Request URL	http://<server>/cgi-bin/storageDevice.cgi?action=getDeviceAllInfo		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.250/cgi-bin/storageDevice.cgi?action=getDeviceAllInfo			

Response Params (key=value format in body)				
Name	Type	R/O	Description	
list	object	R	Return object	
+info	object[]	R	Object array of the storage device information	
++Detail	object[]	O	Partition information	
+++IsError	bool	O	Whether the partition is abnormal	false
+++Pointer	uint	O	Partition operation handle	27023434
+++TotalBytes	double	O	Total partition space; unit: Byte	0
+++Type	char[16]	O	Partition type "ReadWrite": Read and write partition "ReadOnly": Read only partition	"ReadWrite"

+++Path	char[128]	O	Partition name	"/mnt/dvr/sda0"
+++UsedBytes	double	O	Occupied partition space; unit: Byte	
++Name	char[32]	R	Device name	"/dev/sda"
++State	char[4]	R	Device status "Error": Failed to get the device status "Initializing": Getting the device status "Success": Successfully get the device status	"Success"

Response Example

```

list.info[0].Detail[0].IsError=false
list.info[0].Detail[0].Pointer=27023434
list.info[0].Detail[0].TotalBytes=0
list.info[0].Detail[0].Type=ReadWrite
list.info[0].Detail[0].Path=/mnt/dvr/sda0
list.info[0].Detail[0].UsedBytes=0
list.info[0].Name=/dev/sda
list.info[0].State=Success

```

6.1.4 Get Storage Capability

Table 6-1

Syntax	http://<server>/cgi-bin/storage.cgi?action=getCaps
Method	GET
Description	Get storage capabilities.
Example	http://192.168.1.250/cgi-bin/storage.cgi?action=getCaps
Success	caps.RedundantDisk.Support=false
Return	caps.SupportRemoteLimit=true
Comment	—

6.1.5 Format Camera SD-Card

Table 6-2

Syntax	http://<server>/cgi-bin/storageDevice.cgi?action=setDevice&type=FormatPartition&path=<path> [&<paramName>=<paramValue>...]		
Method	GET		
Description	Format camera SD card.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
type	string	R	operation
path	string	O	disk part path
formattype	Array<object>	O	disk part information
+fs	string	R	File system type
Example	http://192.168.1.250/cgi-bin/storageDevice.cgi?action=setDevice&type=FormatPartition&pa		

	th= /dev/sda&formattype[0].fs=fat32
Success Return	OK
Comment	If set successfully, return true, else return false. path : The stringValue is got from cgi API " <i>Get storage device information</i> " (/cgi-bin/storageDevice.cgi?action=getDeviceAllInfo). stringValue is list[0].Name .

6.1.6 [Config] Hard Disk Recording Type

Config Data Params				
Name	Type	R/O	Description	Example
SupportDiskRecordType	object	O	Supported disk recording type info.	
+DiskRecordType	char[32]	O	Disk recording mode "SMR", "CMR"	"SMR"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example
http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=SupportDiskRecordType
Get Config Response Example
table.SupportDiskType.DiskRecordType =SMR

Set Config Request Example
http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&SupportDiskRecordType.DiskRecordType=SMR
Set Config Response Example
OK

6.1.7 Getting Disk Information

Request URL	http://<server>/cgi-bin/api/StorageDeviceManager/getDeviceInfos			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
volume	char[32]	R	Volume type enumchar[32]{ "PhysicalVolume": physical volume (includes physical disk in RAID and VG, and includes USB flash drive). "IndividualPhysicalVolume": Individual physical volume (does not participate in RAID, virtual, and clone groups).	"PhysicalVolume"

		<p>"RaidVolume": Raid volume "IndividualRaidVolume": individual RAID volume (does not participate in virtual and other groups) "VolumeGroup": VG virtual volume group "iSCSI": iSCSI volume "GlobalSpareVolume": global hot standby volume "NAS": NAS volume (Includes FTP, SAMBA, NFS) } Keep consistent with the definition of Device Information Volume type value is "", get the information of all volumes, including IndividualPhysical、IndividualRaid、 VolumeGroup</p>	
Request Example			
<pre>{ "volume": "PhysicalVolume" }</pre>			

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
device	object[256]	O	Disk Information	[, ...,]
+Name	char[32]	O	Device name	"/dev/sda"
+Media	char[]	O	Media type "CDROM": disk "DISK": hard drive "FLASH": SD card "FROCK": frock "UNKNOWN": unknown	"CDROM"
+BUS	char[8]	O	Bus type, supporting "ATA", "SATA", "USB", "SDIO", "SCSI", "SAS" and more.	"ATA"
+PhysicalID	uint	O	Physical number of the similar device storage port, starting from 1.	1
+LogicalID	uint	O	Logic number of the similar device storage port	0
+Capacity	double	O	Total capacity, Unit: Byte	314537952
+Volume	enumchar[32]	O	Volume type enumchar[32]{ "PhysicalVolume": independent physical volume/global hot standby }	"RaidVolume"

			<p>"RaidVolume": Raid volume (Includes sub-disk of the soft Raid and local hot standby)</p> <p>"VolumeGroup": VG virtual volume group (once VG is composed of single disk or Raid, then the result will be VG)</p> <p>"iSCSI": iSCSI volume</p>	
+State	enumchar[32]	O	<p>Physical hard disk status information</p> <p>enumchar[32]{</p> <p>"Error": general faults.</p> <p>Some partition still can be used.</p> <p>"Fatal": serious faults, all of the partitions cannot be used (new DVR fault type)</p> <p>"Offline": offline status of physical disks (not used)</p> <p>"Running": the operating status of the physical disk</p> <p>"RaidSub": the disk now is a single disk and it was a Raid sub disk before. It is likely to automatically add Raid after the device is restarted.</p> <p>/*(the four status are used in single disk)*/</p> <p>"Active": RAID active</p> <p>"Sync": RAID sync</p> <p>"Spare": RAID hot standby (local)</p> <p>"Faulty": RAID invalid</p> <p>"Rebuilding": RAID rebuild</p> <p>"Removed": RAID removed (not used)</p> <p>"WriteError": RAID write error</p> <p>"WantReplacement": RAID need to be replaces</p> <p>"Replacement": RAID is replacement</p> <p>/*(the 9 status are used in RAID sub disk)*/</p> <p>"GlobalSpare": global hot</p>	"Running"

			<pre> standby /*(the status is used in global hot standby disk)*/ "SnapshotParent": snapshot parent "SnapshotChild": snapshot child "VolumeCloneParent": volume clone parent "VolumeCloneChild": volume clone child /*(the 4 status are used in snapshot/volume clone disk)*/ } </pre>	
+Parent	char[32]	O	Name of the storage physical device or name of the parent storage group of the virtual device. It might be VG or RAID.	"/dev/md0"
+Type	enumchar[32]	O	Disk feature (read and write, read-only) enumchar[32]{ "ReadWrite" "ReadOnly" }	"ReadWrite"
+Module	char[32]	O	Device module	"HCT7210SLA360"
+SerialNo	char[32]	O	Device serial number	"STH607MGAS"
+Temperature	float	O	Single disk temperature (unit: °C)	38.5
+Firmware	char[32]	O	Firmware version	"ST6OA31E"
+Partitions	object[]	O	Partition information Each partition has a WorkDirectory. When the disk has fault or is not formatted, the field does not exist.	
++Name	char[32]	O	Partition name	"/dev/sda0"
++Start	double	O	Start to offset. Unit: Byte	0
++Total	double	O	Total capacity. Unit: Byte	2000000000
++Remain	double	O	The remained capacity. Unit: Byte	1000000000
++MountOn	char[256]	O	Mount point. The App use this directory name to access files.	"/mnt/dvr/idea0"
++FileSystem	char[16]	O	File system, such as "DHFS", "EXT3", "EXT4",	"VFAT"

			"XFS", "VFAT", "UNKNOWN" and more. Empty means it is not formatted.	
++IsSupportFs	bool	O	Whether the device supports the current file system.	"ture"
++Status	enumchar[32]	O	Partition status information enumchar[32]{ "LvAvailable": LV available status "LvNotAvailable": LV unavailable status}	"LvAvailable"
++Group	char[32]	O	Partition working group	"ReadWrite2"
++IsError	bool	O	Whether the file system of the working directory fails	false
+Raid	object	O	RAID information, and it is only valid to RAID volume.	
++Members	char[][16]	O	RAID member, including sub disk and local hot standby disk.	["/dev/sda", "/dev/sdb", "/dev/sdc"]
++MemberInfos	object[]	O	RAID member information one-dimensional array, and the index is corresponding to Members.	
+++ID	uint16	O	Disk number, and it is used to describe the slot of the disk on the disk cabinet.	1
+++Spare	bool	O	Whether it is local hot standby. True: local hot standby False: RAID sub disk	true
++Level	uint8	O	RAID Level	5
++RaidDevices	uint16	O	Number of RAID devices	3
++TotalDevices	uint16	O	Total number of RAID devices	4
++State	char[][16]	O	RAID status, and it is a combination of various status. RAID sub disk status is achieved by State. Supports status "Active" "Inactive" "Clean" "Failed" "Degraded" "Recovering" "Resyncing" "Reshaping" "Checking" "NotStarted" "Readonly" "DataCorrupt" "CreateVG" RAID disk is	["Active", "Degraded"]

			being created to VG	
++ActiveDevices	uint16	O	Number of active devices	3
++WorkingDevices	uint16	O	Number of working devices	3
++FailedDevices	uint16	O	Number of failed devices	3
++SpareDevices	uint16	O	Number of hot standby devices	3
++RecoverPercent	double	O	Recover percent. Its is valid when "Recovering" or "Resyncing" is in RAID status. Value: [0, 100]	12.3
++RecoverMBps	double	O	Recover speed. Its is valid when "Recovering" or "Resyncing" is in RAID status. Unit: Mbps	29.531
++RecoverTimeRemaining	double	O	Remaining time. When in RAID status, "Recovering" or "Resyncing" is valid. Unit: Minute (s)	30.0
++AliasName	char[24]	O	RAID alias, UTF-8 code.	"Image storage RAID"
++Sync	enumint8	O	Sync mode. I/O resources distributing strategy enumint8{ 0: Self-adaptive (The default value is) 1: I / O is prioritized to RAID synchronization. 2: I / O is prioritized to hard drive to write data. 3: Balance }	0
+ISCSI	object	O	ISCSI information. It is only valid for ISCSI disk.	
++Name	char[16]	O	iSCSI name. It is the same with the Name in NAS configuration.	"iSCSIA"
+Tank	object	O	Information of the expansion drawer where the disk in (Note: it refers to Cplus expansion drawer, not the disk cabinet) You can get the information of the through storage.getTankInfo	
++Level	int	O	Control module is level 0.	1
++Slot	uint16	O	Board number on the related cabinet. (starting	0

			from 0). Each cabinet can be inserted on many expansion boards and each expansion drawer (such as Raid card) can cascade many cabinets.	
++TankNo	uint16	O	Expansion port number of the expansion drawer at the same level (starting from 0). There may have many ports on the expansion board (such as SAS)	1
+Slot	int	O	Slot of the disk.	0
+PowerMode	enumchar[32]	O	Hard drive power status enumchar[32]{ "None": unknown status "Active": active status "StandBy": standby status "Idle": idle status }	"StandBy"
+PreDiskCheck	enumchar[32]	O	Hard drive pre-check status (EVS optional field, and uses with disk pre-check function) enumchar[32]{ "Good": read speed of the hard drive can be up to 120, and there are few mistakes in smart information. Nothing else is wrong. "Warn": there are several mistake records in cmd information and there are mistake records in smart information. "Error": there are several mistake records in cmd information and there are mistake records in smart information. There are records of bad sectors. "Willfail": hard drive speed is low, under 64 M. There are several mistake records in cmd information and there are mistake records in smart information. There	"Good"

			are records of bad sectors. "Fail": the hard drive failed to return. "None": unknown status "Becheck": checking status "Checkfail": failed to check }	
+OpState	enumint	O	enumint{ 0: Normal working status 1: Standby status 2: Waiting to be formatted 3: Formatting... 4: Waiting to be fragmentated 5: Fragmenting... 6: Waiting to create RAID 7: Creating RAID... 8: Waiting to delete RAID 9: Deleting RAID... 10: Waiting to fix file system 11: Fixing... 12: Waiting to pre-check 13: Pre-checking... 14: Configuring hot standby disk... 15: Create storage pool 16: Delete storage pool }	0
+ManuFactory	enumchar[32]	O	Hard drive device manufacturer enumchar[32]{ "Unknown" "WD": WD "SG" : Seagate }	"WD"
+PosLedState	enumint8	O	Status of the hard dive GPS indicator enumint8{ 0: The hard dive GPS indicator is off 1: The hard dive GPS indicator is on }	0
+CmrSize	uint32	O	It refers to CMR hard drive, and the unit is sector (513 bytes) Otherwise, the field is 0.	0

+MRTType	enumint	O	Disk recording method enumint{ 0: CMR 1: PMR 2: SMR }	0
+MediaType	char[32]	O	Disk media type	"SSD"
+DRTTypeMixState	enumchar[32]	O	Hard drive recording method enumchar[32]{ "Single": single record method, "Mix": mixed record method. }	"Mix"

Example

```
{
  "device": {
    "Name" : "/dev/sda",
    "Media" : "CDROM",
    "BUS" : "ATA",
    "PhysiINDo" : 1,
    "LogiINDo" : 0,
    "Capacity" : 314537952,
    "Volume": "RaidVolume",
    "State": "Running",
    "Parent" : "/dev/md0",
    "Type" : "ReadWrite",
    "Module" : "HCT7210SLA360",
    "SerialNo" : "STH607MGAS",
    "Temperature" : 38.5,
    "Firmware" : "ST6OA31E",
    "Partitions" : [ {
      "Name" : "/dev/sda0",
      "Start" : 0,
      "Total" : 20000000000,
      "Remain" : 10000000000,
      "MountOn" : "/mnt/dvr/idea0",
      "FileSystem" : "VFAT",
      "IsSupportFs" : "ture",
      "Status" : "LvAvailable",
      "Group" : "ReadWrite2",
      "IsError" : false
    },...,{}],
    "Raid" : {
      "Members" : ["/dev/sda", "/dev/sdb", "/dev/sdc"],
      "MemberInfos" : [
        {
          "ID" : 1,

```

```

        "Spare" : true,
    }, ..., {}],
    "Level" : 5,
    "RaidDevices" : 3,
    "TotalDevices" : 4,
    "State" : ["Active", "Degraded"],
    "ActiveDevices" : 3,
    "WorkingDevices" : 3,
    "FailedDevices" : 3,
    "SpareDevices" : 3,
    "RecoverPercent" : 12.3,
    "RecoverMBps" : 29.531,
    "RecoverTimeRemain" : 30.0,
    "AliasName" : "image storage RAID"
    "Sync" : 0
},//end of Raid
"ISCSI" : {
    "Name" : "iSCSIA"
}, // end of ISCSI
"Tank" : {
    "Level" : 1,
    "Slot" : 0,
    "TankNo" : 1
}, // end of Tank
"Slot" : 0,
"PowerMode" : "StandBy",
"PreDiskCheck" : "Good",
"OpState" : 0
"ManuFactory": "WD",
"PosLedState": 0,
"CmrSize": 0,
"MRType": 0,
"MediaType": "SSD",
"DRTypemixState": "Mix"
}
}
}

```

6.2 NAS

6.2.1 [Config] NAS Information

- Get NAS config

Table 6-3

Syntax	<a href="http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=NAS">http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=NAS
--------	---

Method	GET
Description	Get all the directories on the NAS server.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=NAS
Success Return	table.NAS[0].Name="FTP1" table.NAS[0].Enable=true table.NAS[0].Protocol="FTP" table.NAS[0].Address="www.ttt.com" table.NAS[0].Port=21 table.NAS[0].UserName="anonymity" table.NAS[0].Password="none" table.NAS[0].Directory="share"
Comment	—

- Set NAS config

Table 6-4

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set NAS config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&NAS[0].Name=nas01&NAS[0].Enable=true
Success Return	OK
Comment	Parameters in URL: In table below, Head=NAS[index] index: The index of the NAS Server

Appendix:

ParamName	ParamValue type	Description
Head.Name	string	NAS name.
Head.Enable	bool	Enable/Disable the NAS.
Head.Protocol	string	The range is {"FTP", "SFTP", "SMB", "NFS", "Cloud", "ISCSI" }
Head.Address	string	The IP address or host name.
Head.Port	integer	NAS port.
Head.UserName	string	NAS username.
Head.Password	string	NAS password.
Head.Directory	string	Directory name.

6.3 Storage Point

6.3.1 [Config] Record Storage Point

- Get record storage point config

Table 6-5

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=RecordStoragePoint
Method	GET
Description	Get record storage point config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=RecordStoragePoint
Success Return	table.RecordStoragePoint [0].TimingRecord.Local=1 table.RecordStoragePoint [0].TimingRecord.Redundant=Redundant table.RecordStoragePoint [0].TimingRecord.Remote=FTP table.RecordStoragePoint [0].TimingRecord.AutoSync=false table.RecordStoragePoint [0].TimingRecord.AutoSyncRange=0 table.RecordStoragePoint [0].TimingRecord.LocalForEmergency=false table.RecordStoragePoint [0].TimingRecord.CompressBefore=15
Comment	—

- Set record storage point config

Table 6-6

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set record storage point config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&RecordStoragePoint[0].TimingRecord.Local=local
Success Return	OK
Comment	Parameters in URL: In table below, ch : integer, array index starts from 0, which means video channel>equals to video channel index -1, and so 0 means channel 1). recType : The range is {"TimingRecord", "VideoDetectRecord", "AlarmRecord", "EventRecord", "TimingSnapShot", "VideoDetectSnapShot", "AlarmSnapShot", "EventSnapShot"}

Appendix:

ParamName	ParamValue type	Description
RecordStoragePoint[ch].[recType].Local	integer/boolean	Local directory number, count from 1. For some reason, The front-end devices use Boolean type. True for SD card, false for no SD card.
RecordStoragePoint[ch].[recType].Redundant	string	Redundant directory name.
RecordStoragePoint[ch].[recType].Remote	string	Remote directory name.

ParamName	ParamValue type	Description
RecordStoragePoint[ch].[recType].AutoSync	bool	When remote directory recovers, auto synchronize local directory to remote directory or not.
RecordStoragePoint[ch].[recType].AutoSyncRange	integer	From the remote directory recovering time, how long the data needs to be synchronized. The unit is hour. If it is 0, all the data needs to be synchronized.
RecordStoragePoint[ch].[recType].LocalForEmergency	bool	When the remote directory is unusable, save the data the local directory or not.
RecordStoragePoint[ch].[recType].CompressBefore	integer	The days' data which will be compressed.

6.3.2 [Config] Storage Group

- Get storage group config

Table 6-7

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=StorageGroup
Method	GET
Description	Get storage group config.
Example	http://192.168.1.168/cgi-bin/configManager.cgi?action=getConfig&name=StorageGroup
Success Return	<pre>table.StorageGroup[0].Channels[0].MaxPictures=0 table.StorageGroup[0].FileHoldTime=0 table.StorageGroup[0].Memo=For Reading & Writing Files table.StorageGroup[0].Name=ReadWrite table.StorageGroup[0].OverWrite=true table.StorageGroup[0].PicturePathRule=%y-%M-%d/%c/jpg/%h/%m/%s[%E][%O@%S][%R].jpg table.StorageGroup[0].RecordPathRule=%y-%M-%d/%c/dav/%h/%h.%m.%s-%h.%m.%s[%E][%O@%S][%R].dav table.StorageGroup[1].Channels[0].MaxPictures=0 table.StorageGroup[1].FileHoldTime=0 table.StorageGroup[1].Memo=For FTP Files table.StorageGroup[1].Name=Remote table.StorageGroup[1].OverWrite=true table.StorageGroup[1].PicturePathRule=%y-%M-%d/%c/jpg/%h/%m/%s[%E][%O@%S][%R].jpg table.StorageGroup[1].RecordPathRule=%y-%M-%d/%c/dav/%h/%h.%m.%s-%h.%m.%s[%E][%O@%S][%R].dav</pre>
Comment	—

- Set storage group config

Table 6-8

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set storage group config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&StorageGroup[0].Name=main
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>In table below,</p> <p>Index = Storage Group index</p> <p>ch: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p>

Appendix:

ParamName	ParamValue type	Description
StorageGroup[Index].Name	string	Storage group name.
StorageGroup[Index].Memo	string	Storage group memo.
StorageGroup[Index].FileHoldTime	integer	How many days the file will hold.
StorageGroup[Index].OverWrite	bool	Over write or not when there is not enough storage.
StorageGroup[Index].Channels[ch].MaxPictures	Integer	The max pictures beyond which the old pictures will be over written. If it is 0, the old pictures will be not over written.
StorageGroup[Index].Channels[ch].Path	string	The channel path.

6.4 SD Encrypt

6.4.1 Encrypt SD Card

Table 6-9

Syntax	http://<server>/cgi-bin/SDEncrypt.cgi?action=encrypt&deviceName=<deviceName>&password=<password>
Method	GET
Description	SD encryption operation.
Example	http://192.168.1.250/cgi-bin/SDEncrypt.cgi?action=encrypt&deviceName=/dev/mmc0&password=123456
Success Return	OK
Comment	<p>If set successfully, return true, else return false.</p> <p>deviceName: The stringValue is got from cgi API Get storage device information (cgi-bin/storageDevice.cgi?action=getDeviceAllInfo).</p>

password: The stringValue Card is Encrypted

6.4.2 Decrypt SD Card

Table 6-10

Syntax	<code>http://<server>/cgi-bin/SDEncrypt.cgi?action=decrypt&deviceName=<deviceName>&password=<password></code>
Method	GET
Description	SD decrypt operation.
Example	<code>http://192.168.1.250/cgi-bin/SDEncrypt.cgi?action=decrypt&deviceName=/dev/mmc0&password=123456</code>
Success Return	OK
Comment	If set successfully, return true, else return false. deviceName : The stringValue is got from cgi API Get storage device information (cgi-bin/storageDevice.cgi?action=getDeviceAllInfo). password : The stringValue ard is Encrypted

6.4.3 Clear SD Card Password

Table 6-11

Syntax	<code>http://<server>/cgi-bin/SDEncrypt.cgi?action=clearPassword&deviceName=<deviceName>&password=<password></code>
Method	GET
Description	SD clear password operation.
Example	<code>http://192.168.1.250/cgi-bin/SDEncrypt.cgi?action=clearPassword&deviceName=/dev/mmc0&password=123456</code>
Success Return	OK
Comment	If set successfully, return true, else return false. deviceName : The stringValue is got from cgi API Get storage device information (cgi-bin/storageDevice.cgi?action=getDeviceAllInfo). password : The stringValue ard is Encrypted

6.4.4 Modify SD Card Password

Table 6-12

Syntax	<code>http://<server>/cgi-bin/SDEncrypt.cgi?action=modifyPassword&deviceName=<device Name>&password=<password>&oldPassword=<oldPassword></code>
Method	GET
Description	SD change password operation.
Example	<code>http://192.168.1.250/cgi-bin/SDEncrypt.cgi?action=clearPassword&deviceName=/dev/mmc0&password=123456&oldPassword=654321</code>

Success Return	OK
Comment	If set successfully, return true, else return false. deviceName: The stringValue is got from cgi API Get storage device information (cgi-bin/storageDevice.cgi?action=getDeviceAllInfo). password: The new stringValue Card is Encrypted oldPassword: The old stringValue Card is Encrypted

6.4.5 Get SD Card Operate Error Policy

Table 6-13

Syntax	http://<server>/cgi-bin/SDEncrypt.cgi?action=getOperateErrorPolicy&deviceName=<deviceName>&operate=<operate>
Method	GET
Description	When decrypt, clearPassword, modifyPassword failed, get the error info, leftTimes and left time.
Example	http://192.168.1.250/cgi-bin/SDEncrypt.cgi?action=getOperateErrorPolicy&deviceName=/dev/mmc0&operate=decrypt
Success Return	policy.leftTimes=5 policy.lockSeconds=30
Comment	If set successfully, return true, else return false. deviceName: the stringValue is got from cgi API Get storage device information (cgi-bin/storageDevice.cgi?action=getDeviceAllInfo). Operate: operate type, can be: decrypt, modifyPassword and clearPassword leftTimes : remain operate times, max is 5 lockSeconds : lock operate time, unit is seconds, max is 30

6.4.6 [Config] Storage Health Alarm Settings

- Get StorageHealthAlarm config

Table 6-14

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=StorageHealthAlarm
Method	GET
Description	Get SD health info alarm config.
Example	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=StorageHealthAlarm
Success Return	table.StorageHealthAlarm.Enable=true table.StorageHealthAlarm.LowerLimit=10 table.StorageHealthAlarm.EventHandler= (output of EventHandler is described in GetEventHandler)
Comment	—

- Set StorageHealthAlarm config

Table 6-15

Syntax	<code>http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]</code>
Method	GET
Description	Set SD health info alarm config.
Example	<code>http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&StorageHealthAlarm.Enable=true</code>
Success Return	OK
Comment	—

7. Display API

7.1 GUI

7.1.1 [Config] GUISet

- Get GUISet config

Table 7-1

Syntax	<code>http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=GUISet</code>
Method	GET
Description	Get the GUI settings. Every video out screen has a group setting.
Example	<code>http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=GUISet</code>
Success Return	<code>table.GUISet[index].WindowAlpha=128</code> <code>table.GUISet[index].TimeTitleEnable=true</code> <code>table.GUISet[index].TimeTitlePos[0]=0</code> <code>table.GUISet[index].TimeTitlePos[1]=0</code> <code>table.GUISet[index].TimeTitlePos[2]=8191</code> <code>table.GUISet[index].TimeTitlePos[3]=8191</code> <code>table.GUISet[index].MenuShowOption=0</code> <code>table.GUISet[index].MenuAutoHideTime=10</code> <code>table.GUISet[index].AutoLogout=10</code> <code>table.GUISet[index].ChannelTitleShowEnable=true</code> <code>table.GUISet[index].ChannelTitlePos[0]=0</code> <code>table.GUISet[index].ChannelTitlePos[1]=0</code> <code>table.GUISet[index].ChannelTitlePos[2]=8191</code> <code>table.GUISet[index].ChannelTitlePos[3]=8191</code> <code>table.GUISet[index].AutoGuideEnable=true</code> ...
Comment	Parameters in Response : <code>index</code> : the array index which starts from 0.

- Set GUISet config

Table 7-2

Syntax	<code>http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue> [&<paramName>=<paramValue> ...]</code>
Method	GET
Description	Set the GUI settings. Every video out screen has a group setting.
Example	<code>http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&GUISet[0].WindowAlpha=192&GUISet[0].TimeTitleEnable=false&GUISet[0].MenuShowOption=1</code>

Success Return	OK
Comment	<p>Parameters in URL:</p> <p>The paramName and paramValue are in the table below.</p> <p>in table below,</p> <p>index : the array index which starts from 0.</p>

Appendix:

ParamName	ParamValue type	Description
GUISet[index].WindowAlpha	integer	Diaphaneity of the window background.
GUISet[index].TimeTitleEnable	bool	Show the time title or not.
GUISet[index].TimeTitlePos[0]	integer	The position of the time title.
GUISet[index].TimeTitlePos[1]	integer	
GUISet[index].TimeTitlePos[2]	integer	
GUISet[index].TimeTitlePos[3]	integer	
GUISet[index].MenuShowOption	integer	0: Show the directory. 1: Hide the directory. 2: Timing-hide the directory.
GUISet[index].MenuAutoHideTime	integer	How many seconds to hide the directory.
GUISet[index].AutoLogout	integer	How many minutes to auto logout. The range is [0-120]. 0 expresses not logout.
GUISet[index].ChannelTitleShowEnable	bool	Show the channel title or not.
GUISet[index].ChannelTitlePos[0]	integer	The position of the channel title.
GUISet[index].ChannelTitlePos[1]	integer	
GUISet[index].ChannelTitlePos[2]	integer	
GUISet[index].ChannelTitlePos[3]	integer	
GUISet[index].AutoGuideEnable	bool	Auto guide or not when startup.

7.2 Split Screen

7.2.1 Split Screen Mode

- Get split screen mode

Table 7-3

Syntax	http://<server>/cgi-bin/split.cgi?action=getMode&channel=<ChannelNo>
Method	GET
Description	Get the split screen mode.
Example	http://192.168.1.250/cgi-bin/split.cgi?action=getMode&channel=1
Success Return	mode=split1 group=4
Comment	Parameters in URL:

ChannelNo: the display screen No. Start from 1 and <= 2.

- Set split screen mode

Table 7-4

Syntax	http://<server>/cgi-bin/split.cgi?action=setMode&channel=< ChannelNo >&mode=< mode >&group=< group >
Method	GET
Description	Set the split screen mode.
Example	http://192.168.1.250/cgi-bin/split.cgi?action=setMode&channel=1&mode=split4&group=1
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>ChannelNo: the display screen No. Start from 1.</p> <p>mode:enum{split1,split2,split4,split6,split8,split9,split12,split16,split20,split25,split36,split64,split144,pip1,pip3, "Free", "CompositeSplit1" / "FitDisplayUnit1", "CompositeSplit1" / "FitDisplayUnit4"};</p> <p>group: the No. of a group which contains certain number channels. For example, if 16 video channels display in split4 Mode which contains 4 video channels on Screen, then there are 4 groups and each group contains 4 video channels.</p>

7.3 Monitor Tour

7.3.1 [Config] Monitor Tour

- Get monitor tour config

Table 7-5

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=MonitorTour					
Method	GET					
Description	Get monitor tour config.					
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=MonitorTour					
Success Return	<table border="0"> <tr> <td>table.MonitorTour[ch].Enable=128</td> </tr> <tr> <td>table.MonitorTour[ch].Interval=true</td> </tr> <tr> <td>table.MonitorTour[ch].Mask.Split1=0,1,5</td> </tr> <tr> <td>table.MonitorTour[ch].Mask.Split8=0,1,5</td> </tr> <tr> <td>table.MonitorTour[ch].Collections=Favortite1, Favortite2...</td> </tr> </table>	table.MonitorTour[ch].Enable=128	table.MonitorTour[ch].Interval=true	table.MonitorTour[ch].Mask.Split1=0,1,5	table.MonitorTour[ch].Mask.Split8=0,1,5	table.MonitorTour[ch].Collections=Favortite1, Favortite2...
table.MonitorTour[ch].Enable=128						
table.MonitorTour[ch].Interval=true						
table.MonitorTour[ch].Mask.Split1=0,1,5						
table.MonitorTour[ch].Mask.Split8=0,1,5						
table.MonitorTour[ch].Collections=Favortite1, Favortite2...						
Comment	—					

- Set moniter tour config

Table 7-6

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&< paramName >=< paramValue >[&< paramName >=< paramValue >...]
Method	GET

Description	Set monitor tour config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&MonitorTour[0].Enable=true
Success Return	OK.
Comment	Parameters in URL: The paramName and paramValue are in the table below.

Appendix:

ParamName	ParamValue type	Description
MonitorTour[ch].Enable	bool	MonitorTour or not.
MonitorTour[ch].Interval	integer	MonitorTour interval.
MonitorTour[ch].Mask.Split1	—	Channel array for split1
MonitorTour[ch].Mask.Split8	—	Channel array for split8
MonitorTour[ch].Collections	—	Split collections

7.3.2 Enable Tour

Table 7-7

Syntax	http://<server>/cgi-bin/split.cgi?action=enableTour&channel=< ChannelNo >&enable=< flag >
Method	GET
Description	Enable tour in every video channel on a screen or not.
Example	http://192.168.1.250/cgi-bin/split.cgi?action=enableTour&channel=1&enable=true
Success Return	OK
Comment	ChannelNo : the display screen No. Start from 1 and <= 2. flag : true or false

7.3.3 [Config] Monitor Collection

- Get monitor collection config

Table 7-8

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=MonitorCollection
Method	GET
Description	Get monitor collection config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=MonitorCollection
Success Return	table.MonitorCollection.collectionname.Mode=Split1 table.MonitorCollection.collectionname.Windows[winno].Enable= true table.MonitorCollection.collectionname.Windows[winno].Device=device1 table.MonitorCollection.collectionname.Windows[winno].VideoChannel=5 table.MonitorCollection.collectionname.Windows[winno].VideoStream=Main table.MonitorCollection.collectionname.Windows[winno].AudioChannel=5

	table.MonitorCollection.collectionname.Windows[winno].AudioStream=Main ...
Comment	Parameters in Response : winno : integer, the array index which equals to the window index in a screen and starts from 0.

- Set monitor collection config

Table 7-9

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&< paramName >=< paramValue >[&< paramName >=< paramValue >...]
Method	GET
Description	Set monitor collection config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&MonitorCollection.Favorite1.Mode=split4&MonitorCollection.Favorite1.Windows[1].Enable=true&MonitorCollection.Favorite1.Windows[1].VideoChannel=2
Success Return	OK
Comment	<p>Parameters in URL: The paramName and paramValue are in the table below. In table below: Collect=MonitorCollection.collectionname. collectionname: can be any name. winno: integer, the array index which equals to the window index in a screen and starts from 0.</p>

Appendix:

ParamName	ParamValue type	Description
Collect .Mode	string	The range is the same as <u>SetSplitMode</u> .
Collect .Windows[winno].Enable	bool	Enable the window or not.
Collect .Windows[winno].Device	string	The device Id.
Collect .Windows[winno].VideoChannel	integer	The video channel.
Collect .Windows[winno].VideoStream	string	The range is {"Main", "Extra1", "Extra2", "Extra3", "Auto"}.
Collect .Windows[winno].AudioChannel	integer	The audio channel.
Collect .Windows[winno].AudioStream	string	The range is {"Main", "Extra1", "Extra2", "Extra3", "Auto"}.

8.PTZ API

8.1 PTZ

8.1.1 [Config] PTZ Config

- Get PTZ config

Table 8-1

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=Ptz
Method	GET
Description	Get PTZ config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=Ptz
Success Return	table.Ptz[port].Address=8 table.Ptz[port].Attribute[0]=115200 table.Ptz[port].Attribute[1]=8 table.Ptz[port].Attribute[2]=Even table.Ptz[port].Attribute[3]=1 table.Ptz[port].Homing[0]=0 table.Ptz[port].Homing[1]=30 table.Ptz[port].NumberInMatrixs=0 table.Ptz[port].ProtocolName=NONE
Comment	Parameters in Response: port is PTZ port index, start from 0.

- Set PTZ config

Table 8-2

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue> [&<paramName>=<paramValue> ...]
Method	GET
Description	Set PTZ config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&Ptz[0].Address=192.168.0.1&Ptz[0].Attribute[0]=9600
Success Return	OK
Comment	port in below ParamName is PTZ port index, start from 0.

Appendix:

ParamName	ParamValue type	Description
Ptz[port].Address	integer	Range is [0—255]. Device address, if there are more than one device

ParamName	ParamValue type	Description
		connected to this port, distinguish them by this address.
Ptz[<i>port</i>].Attribute[0]	integer	The baud rate. Range is {1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200}.
Ptz[<i>port</i>].Attribute[1]	integer	Range is {4, 5, 6, 7, 8}. Data bit.
Ptz[<i>port</i>].Attribute[2]	string	Range is {Even, Mark, None, Odd, Space}. Parity verification mode.
Ptz[<i>port</i>].Attribute[3]	float	Range is {1, 1.5, 2}. Stop bit.
Ptz[<i>port</i>].Homing[0]	integer	Range is {-1,0—255} -1: homing is not enabled. [0—255]: preset point number
Ptz[<i>port</i>].Homing[1]	integer	Range is [0—65535]. No operation timeout, unit is seconds. After no operation timeout, PTZ go to preset point set in Ptz[<i>port</i>].Homing[0].
Ptz[<i>port</i>].ProtocolName	string	PTZ protocol name depends on PTZ capability. Refer to GetProtocolList to get the protocol list.
Ptz[<i>port</i>].ControlDelayTime	integer	PTZ control delay time, unit: second
Ptz[<i>port</i>].ControlPriority	string	PTZ control priority, the range is "RS485", "Net". default : "Net"

8.1.2 Get PTZ Protocol List

Table 8-3

Syntax	http://<server>/cgi-bin/ptz.cgi?action=getProtocolList[&channel=< <i>ChannelNo</i> >]
Method	GET
Description	Get the protocol list that PTZ can support. Unsupported now.
Example	http://192.168.1.250/cgi-bin/ptz.cgi?action=getProtocolList&channel=1
Success Return	info.RS[0]=Pelco info.RS[1]=DH-SD1 info.Coaxial[0]=HD-CVI info.Coaxial[1]=HD-CVI2.0
Comment	Response contains all support PTZ protocols of the server. <i>ChannelNo</i> : integer, video channel index which starts from 1.

8.1.3 Get PTZ Capability of Current Protocol

Table 8-4

Syntax	http://<server>/cgi-bin/ptz.cgi?action=getCurrentProtocolCaps[&channel=< <i>ChannelNo</i> >]
Method	GET
Description	Get PTZ channel protocol capabilities.

Example	<code>http://192.168.1.250/cgi-bin/ptz.cgi?action=getCurrentProtocolCaps&channel=1</code>
Success Return	<pre> caps.AlarmLen=0 caps.AuxMax=8 caps.AuxMin=1 caps.CamAddrMax=255 caps.CamAddrMin=1 caps.Flip=false caps.Focus=false caps.Interval=200 caps.Iris=false caps.Menu=false caps.MonAddrMax=255 caps.MonAddrMin=0 caps.Name=DH-SD1 caps.Pan=false caps.PanSpeedMax=255 caps.PanSpeedMin=1 caps.PatternMax=5 caps.PatternMin=1 caps.PresetMax=80 caps.PresetMin=1 caps.Tile=false caps.TileSpeedMax=255 caps.TileSpeedMin=1 caps.TourMax=7 caps.TourMin=0 caps.Type=1 caps.Zoom=false caps.PtzMotionRange.HorizontalAngle[0]=0 caps.PtzMotionRange.HorizontalAngle[1]=360 caps.PtzMotionRange.VerticalAngle[0]=-20 caps.PtzMotionRange.VerticalAngle[1]=90 caps.ZoomMax=30 caps.ZoomMin=1 </pre>
Comment	<p>Parameters in URL:</p> <p>ChannelNo: PTZ channel index which starts from 1.</p>

Appendix:

Field in response	Description
AlarmLen	Alarm length in protocol.
AuxMax	Maximum/Minimum number for auxiliary functions.
AuxMin	Maximum/Minimum number for auxiliary functions.
CamAddrMax	Maximum/Minimum channel address.
CamAddrMin	Maximum/Minimum channel address.
Flip	True or false, support picture flip or not.
Focus	True or false, support focus or not.
Iris	True or false, support Iris adjusts or not.

Field in response	Description
Menu	True or false, support internal menu of the PTZ or not.
MonAddrMax	Maximum/Minimum monitor address.
MonAddrMin	
Name	Name of the operation protocol.
Pan	True or false, support pan or not.
PanSpeedMax	Maximum/Minimum pan speed.
PanSpeedMin	
PatternMax	Maximum/Minimum pattern path number.
PatternMin	
PresetMax	Maximum/Minimum preset point number.
PresetMin	
Tile	True or false, support tilt or not.
Zoom	True or false, support zoom or not.
TileSpeedMax	Maximum/Minimum tile speed.
TileSpeedMin	
TourMax	Maximum/Minimum tour path number.
TourMin	
Type	Type of PTZ protocol.
PtzMotionRange.HorizontalAngle	Horizontal angle range, [0] for minimum angle, [1] for maximum angle it only when Pan was true
PtzMotionRange.VerticalAngle	Vertical angle range, [0] for minimum angle, [1] for maximum angle it only when Tile was true
ZoomMax	Maximum/Minimum Zoom.
ZoomMin	it only when Zoom was true

8.1.4 Get PTZ Status

Table 8-5

Syntax	http://<server>/cgi-bin/ptz.cgi?action=getStatus[&channel=<ChannelNo>]
Method	GET
Description	Get PTZ status.
Example	http://192.168.1.250/cgi-bin/ptz.cgi?action=getStatus&channel=1
Success Return	status.UTC=6538920 status.MoveStatus=Idle status.ZoomStatus=Idle status.PresetID=10 status.Position=120,12,2
Comment	ChannelNo: integer, video channel index which starts from 1.

8.1.5 PTZ Control

- PTZ Basic Movement

Start moving the PTZ.

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	The PTZ channel index; starting from 1	1
code	char[16]	R	See the following table for the operation codes for PTZ movement.	"Up"
arg1	int	O	Operation parameter 1: See the following table for the meaning of the operation code.	0
arg2	int	O	Operation parameter 2: See the following table for the meaning of the operation code.	1
arg3	int	O	Operation parameter 3: See the following table for the meaning of the operation code.	0

Request Example

http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=Up&arg1=0&arg2=1&arg3=0

Response Params (OK in body)
Response Example
OK

Stop moving the PTZ.

Request URL	http://<server>/cgi-bin/ptz.cgi?action=stop			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	The PTZ channel index; starting from 1	1
code	char[16]	R	See the following table for the operation codes for PTZ movement.	"Up"
arg1	int	O	Operation Parameter 1, reserved.	0
arg2	int	O	Operation Parameter 2, reserved.	0
arg3	int	O	Operation Parameter 3, reserved.	0

Request Example

http://192.168.1.250/cgi-bin/ptz.cgi?action=stop&code=Up&channel=1&arg1=0&arg2=0&arg3=0

Response Params (OK in body)
Response Example
OK

Appendix: Operation codes for PTZ movement and the parameters.

Code	Code description	arg1	arg2	arg3
Up	Move up	0	Vertical motion speed; range: [1–8]	0
Down	Move down	0	Vertical motion speed; range: [1–8]	0
Left	Move left	0	Horizontal motion speed; range: [1–8]	0

Code	Code description	arg1	arg2	arg3
Right	Move right	0	Horizontal motion speed; range: [1–8]	0
LeftUp	Move in an upper-left direction	Vertical motion speed; range: [1–8]	Horizontal motion speed; range: [1–8]	0
RightUp	Move in an upper-right direction	Vertical motion speed; range: [1–8]	Horizontal motion speed; range: [1–8]	0
LeftDown	Move in an lower-left direction	Vertical motion speed; range: [1–8]	Horizontal motion speed; range: [1–8]	0
RightDown	Move in an lower-right direction	Vertical motion speed; range: [1–8]	Horizontal motion speed; range: [1–8]	0
ZoomWide	Zoom in	0	0	0
ZoomTele	Zoom out	0	0	0
FocusNear	Focus (near-field)	0	0	0
FocusFar	Focus (far-field)	0	0	0
IrisLarge	Increase the aperture	0	0	0
IrisSmall	Decrease the aperture	0	0	0

- PTZ Continuously Moving

Start continuously moving the PTZ

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=Continuously			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	The PTZ channel index; starting from 1	1
code	char[16]	R	PTZ continuously moving operation code, must be "Continuously".	"Continuously"
arg1	int	O	See the table below for the motion direction and step length.	5
arg2	int	O	See the table below for the motion direction and step length.	5
arg3	int	O	Zooming speed; range: [-100–100]	5
arg4	int	O	Overtime period; unit: s (maximum 3600 s). If the PTZ does not receive the stop command before the preset overtime period, it will stop moving automatically.	60
Request Example				
http://192.168.1.250/cgi-bin/ptz.cgi?action=start&code=Continuously&channel=1&arg1=5&arg2=5&arg3=5&arg4=60				

Response Params (OK in body)
Response Example
OK

Appendix: Direction of the continuous movement of the PTZ and the step length parameters.

Move description	arg1	arg2
------------------	------	------

Move description	arg1	arg2
Continuously move left	< -4	0
Continuously move right	> 4	0
Continuously move up	0	> 4
Continuously move down	0	< -4
Continuously move in an upper-left direction	< -4	> 4
Continuously move in an upper-right direction	> 4	> 4
Continuously move in a lower-left direction	< -4	< -4
Continuously move in a lower-right direction	> 4	< -4

Stop continuously moving the PTZ

Request URL	http://<server>/cgi-bin/ptz.cgi?action=stop&code=Continuously			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	The PTZ channel index; starting from 1	1
code	char[16]	R	PTZ continuously moving operation code, must be "Continuously".	"Continuously"
arg1	int	O	Operation Parameter 1, reserved.	0
arg2	int	O	Operation Parameter 2, reserved.	0
arg3	int	O	Operation Parameter 3, reserved.	0
arg4	int	O	Operation Parameter 4, reserved.	0
Request Example				
http://192.168.1.250/cgi-bin/ptz.cgi?action=stop&code=Continuously&channel=1&arg1=0&arg2=0&arg3=0&arg4=0				

Response Params (OK in body)
Response Example
OK

- 3D Positioning

The PTZ moves to the specified position [startX, startY], [endX, endY] on the screen through 3D positioning .

Request URL	http://<server>/cgi-bin/ptzBase.cgi?action=moveDirectly			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	The PTZ channel index; starting from 1	1
startPoint	int[2]	R	The start point of the target matrix [startX, startY]; relative coordinate; the value of X and Y is normalized to a number between 0 and 8192.	[7253,2275]
endpoint	int[2]	R	The end point of the target matrix [startX, endX, startY, endY]; relative coordinate; the value of X and Y is normalized to a number between 0 and 8192.	[7893,3034]

		startY]; relative coordinate; the value of X and Y is normalized to a number between 0 and 8192.	
--	--	--	--

Request Example

```
http://192.168.1.250/cgi-bin/ptzBase.cgi?action=moveDirectly&channel=1&startPoint[0]=7253&startPoint[1]=2275&endPoint[0]=7893&endPoint[1]=3034
```

Response Params (OK in body)

Response Example

OK

- Relative PTZ Movement

Relative PTZ movement

Request URL	http://<server>/cgi-bin/ptz.cgi?action=moveRelatively		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	R	The PTZ channel index; starting from 1
arg1	double	O	Relative horizontal motion; normalized to [-1, 1]
arg2	double	O	Relative vertical motion; normalized to [-1, 1]
arg3	double	O	Relative zoom; normalized to [-1, 1]

Request Example

```
http://192.168.1.250/cgi-bin/ptz.cgi?action=moveRelatively&channel=1&arg1=0.1&arg2=0.1&arg3=0.5
```

Response Params (OK in body)

Response Example

OK

- Accurate PTZ Positioning

Accurate PTZ positioning; the actual parameter range depends on the device. For capability details, execute the command mentioned in "8.1.3 Get PTZ Capability of Current Protocol".

Request URL	http://<server>/cgi-bin/ptz.cgi?action=moveAbsolutely		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	R	The PTZ channel index; starting from 1
arg1	double	O	Relative horizontal position; normalized to [-1, 1]; arg1 < 0: Angle = 180.0 × arg1 + 360.0, and the actual range is [180.0, 360.0]; arg1 ≥ 0: Angle = 180.0 × arg1, and the actual range is [0, 180.0]
arg2	double	O	Absolute vertical position; normalized to [-1, 1]; Angle = -180.0 × arg2, the actual range is [-180.0, 180.0]
arg3	double	O	Absolute zoom; normalized to [-1, 1]

Request Example

http://192.168.1.250/cgi-bin/ptz.cgi?action=moveAbsolutely&channel=1&arg1=-0.8&arg2=0.3&arg3=0.5

Response Params (OK in body)

Response Example

OK

8.1.6 Preset

- Getting Preset Information

Get the preset information

Request URL	http://<server>/cgi-bin/ptz.cgi?action=getPresets			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	Video channel number, starting from 1; the default value is 1.	1
Request Example				
http://192.168.1.250/cgi-bin/ptz.cgi?action=getPresets&channel=1				

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
presets	object[]	R	List of preset information	
+Index	int	R	The preset number; starting from 1	1
+Name	char[256]	O	The preset name	"preset1"
+Type	int	O	Preset type 0: Normal preset 1: Preset configured with smart rules 2: Special preset	0
+PresetFunction	char[16]	O	Function List of the Special Presets "VideoBlack": Day/Night Mode (B/W) "VideoColor": Day/Night Mode (Color) "VideoBrightness": Day/Night Mode (Auto)	"VideoBlack" "
+Position	int[3]	O	The coordinate of the preset and zoom; three integers The first parameter is the horizontal coordinate; range: [0,3599], referring to 0° to 359.9° (the number of degree is expanded by 10 times). The second parameter is the vertical coordinate; range: [-1800,1800], referring to -180° to 180° (the number of degree is expanded by 10 times). The third parameter is the expansion parameter; range: [0,128], referring to the zoom range.	[900, -900, 5]

Response Example

presets[0].Index=1

```

presets[0].Name=Preset 1
presets[0].Type=0
presets[0].PresetFunction="VideoBlack"
presets[0].Position=[900, -900, 5]
...

```

- Moving to the Preset

Move to the preset.

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=GotoPreset			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	The PTZ channel index; starting from 1	1
arg1	int	O	Ignore	0
arg2	int	R	The preset number; starting from 1	1
arg3	int	O	Ignore	0
Request Example				
http://192.168.1.108/cgi-bin/ptz.cgi?action=start&code=GotoPreset&channel=1&arg1=0&arg2=1&arg3=0				

Response Params (OK in body)
Response Example
OK

- Configuring Preset

Configure a preset.

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=SetPreset			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	The PTZ channel index; starting from 1	1
arg1	int	O	Ignore	0
arg2	int	R	The preset number; starting from 1	1
arg3	int	O	Ignore	0
Request Example				
http://192.168.1.250/cgi-bin/ptz.cgi?action=start&code=SetPreset&channel=1&arg1=0&arg2=2&arg3=0				

Response Params (OK in body)
Response Example
OK

- Configuring Preset Name

Configure a name for the preset.

Request URL	http://<server>/cgi-bin/ptz.cgi?action=SetPreset			
Method	GET			
Request Params (key=value format in URL)				

Name	Type	R/O	Description	Example
channel	int	R	The PTZ channel index; starting from 1	1
arg1	int	R	The preset number; starting from 1	2
arg2	char[256]	R	The preset name	"preset2"

Request Example

http://192.168.1.250/cgi-bin/ptz.cgi?action=SetPreset&channel=1&arg1=2&arg2=preset2

Response Params (OK in body)

Response Example

OK

- Deleting Preset

Delete a preset.

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=ClearPreset			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	The PTZ channel index; starting from 1	1
arg1	int	O	Ignore	0
arg2	int	R	The preset number; starting from 1	2
arg3	int	O	Ignore	0

Request Example

http://192.168.1.250/cgi-bin/ptz.cgi?action=start&code=ClearPreset&channel=1&arg1=0&arg2=2&arg3=0

Response Params (OK in body)

Response Example

OK

8.1.7 Tour

- Start the tour

Table 8-6

Syntax	http://<server>/cgi-bin/ptz.cgi?action=start&channel=< ch >&code=StartTour&arg1=< index >&arg2=0&arg3=0
Method	GET
Description	Start the tour.
Example	http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=StartTour&arg1=1&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ch is PTZ channel index which starts from 1. Range is [1 — n]. arg1 is the number of tour route

- Stop the tour

Table 8-7

Syntax	<code>http://<server>/cgi-bin/ptz.cgi?action=start&channel=<ch>&code=StopTour&arg1=<index>&arg2=0&arg3=0</code>
Method	GET
Description	Stop the tour.
Example	<code>http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=StopTour&arg1=1&arg2=0&arg3=0</code>
Success Return	OK
Comment	Parameters in URL: ch is PTZ channel index which starts from 1. Range is [1 — n]. arg1 is the number of tour route

- Add tour group

Table 8-8

Syntax	<code>http://<server>/cgi-bin/ptz.cgi?action=setTour&channel=<ch>&arg1=<index>&arg2=<name></code>
Method	GET
Description	Add tour group, including set the tour route and name. Frequently used with add tour preset.
Example	<code>http://192.168.1.250/cgi-bin/ptz.cgi?action=setTour&channel=1&arg1=1&arg2=1</code>
Success Return	OK
Comment	Parameters in URL: ch is PTZ channel index which starts from 1. Range is [1 — n]. arg1 is the number of tour route arg2 is tour name

- Delete tour group

Table 8-9

Syntax	<code>http://<server>/cgi-bin/ptz.cgi?action=start&channel=<ch>&code=ClearTour&arg1=<index>&arg2=0&arg3=0</code>
Method	GET
Description	Delete the tour group with specified number.
Example	<code>http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=ClearTour&arg1=1&arg2=0&arg3=0</code>
Success Return	OK
Comment	Parameters in URL: ch is PTZ channel index which starts from 1. Range is [1 — n]. arg1 is the number of tour group

- Add tour preset

Table 8-10

Syntax	<code>http://<server>/cgi-bin/ptz.cgi?action=start&channel=<ch>&code=AddTour&arg1=<index1>&arg2=<index2>&arg3=0</code>
Method	GET
Description	Add preset to tour group.
Example	<code>http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=AddTour&arg1=1&arg2=2&arg3=0</code>
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>ch is PTZ channel index which starts from 1. Range is [1 — n].</p> <p>arg1 is the number of tour route</p> <p>arg2 is the number of preset</p>

- Delete tour preset

Table 8-11

Syntax	<code>http://<server>/cgi-bin/ptz.cgi?action=start&channel=<ch>&code=DelTour&arg1=<index1>&arg2=<index2>&arg3=0</code>
Method	GET
Description	Delete the specified preset from a particular tour group.
Example	<code>http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=DelTour&arg1=1&arg2=2&arg3=0</code>
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>ch is PTZ channel index which starts from 1. Range is [1 — n].</p> <p>arg1 is the number of tour route</p> <p>arg2 is the number of preset</p>

8.1.8 Scan

- Set left boundary

Table 8-12

Syntax	<code>http://<server>/cgi-bin/ptz.cgi?action=start&channel=<ch>&code=SetLeftLimit&arg1=<index>&arg2=0&arg3=0</code>
Method	GET
Description	Set left boundary.
Example	<code>http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=SetLeftLimit&arg1=1&arg2=0&arg3=0</code>
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>ch is PTZ channel index which starts from 1. Range is [1 — n].</p>

arg1 is the scan number

- Set right boundary

Table 8-13

Syntax	http://<server>/cgi-bin/ptz.cgi?action=start&channel=< ch >&code=SetRightLimit&arg1=< in dex >&arg2=0&arg3=0
Method	GET
Description	Set right boundary.
Example	http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=SetRightLimit&arg1=1&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ch is PTZ channel index which starts from 1. Range is [1 — n]. arg1 is the scan number

- Start scan

Table 8-14

Syntax	http://<server>/cgi-bin/ptz.cgi?action=start&channel=< ch >&code=AutoScanOn&arg1=< in dex >&arg2=0&arg3=0
Method	GET
Description	Start scanning.
Example	http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=AutoScanOn&arg1=1&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ch is PTZ channel index which starts from 1. Range is [1—n]. arg1 is scan number

- Stop scan

Table 8-15

Syntax	http://<server>/cgi-bin/ptz.cgi?action=start&channel=< ch >&code=AutoScanOff&arg1=< in dex >&arg2=0&arg3=0
Method	GET
Description	Stop scanning.
Example	http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=AutoScanOff&arg1=1&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ch is PTZ channel index which starts from 1. Range is [1 — n]. arg1 is scan number

8.1.9 Pattern

- Start pattern record

Table 8-16

Syntax	http://<server>/cgi-bin/ptz.cgi?action=start&channel=< ch >&code=SetPatternBegin&arg1=< index >&arg2=0&arg3=0
Method	GET
Description	Start pattern record.
Example	http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=SetPatternBegin&arg1=1&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ch is PTZ channel index which starts from 1. Range is [1 — n]. arg1 is pattern number

- Stop pattern record

Table 8-17

Syntax	http://<server>/cgi-bin/ptz.cgi?action=start&channel=< ch >&code=SetPatternEnd&arg1=< i ndex >&arg2=0&arg3=0
Method	GET
Description	Stop pattern record.
Example	http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=SetPatternEnd&arg1=1&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ch is PTZ channel index which starts from 1. Range is [1 — n]. arg1 is pattern number

- Start pattern

Table 8-18

Syntax	http://<server>/cgi-bin/ptz.cgi?action=start&channel=< ch >&code=StartPattern&arg1=< ind ex >&arg2=0&arg3=0
Method	GET
Description	Start pattern.
Example	http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=StartPattern&arg1=1&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ch is PTZ channel index which starts from 1. Range is [1 — n]. arg1 is pattern number

- Stop pattern

Table 8-19

Syntax	http://<server>/cgi-bin/ptz.cgi?action=start&channel=< ch >&code=StopPattern&arg1=< ind ex >&arg2=0&arg3=0
Method	GET
Description	Stop pattern.
Example	http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=StopPattern&arg1=1&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ch is PTZ channel index which starts from 1. Range is [1 — n]. arg1 is pattern number

8.1.10 Pan

- Start pan

Table 8-20

Syntax	http://<server>/cgi-bin/ptz.cgi?action=start&channel=< ch >&code=AutoPanOn&arg1=0&arg2=0&arg3=0
Method	GET
Description	Start pan.
Example	http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=AutoPanOn&arg1=0&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ch is PTZ channel index which starts from 1. Range is [1 — n].

- Stop pan

Table 8-21

Syntax	http://<server>/cgi-bin/ptz.cgi?action=start&channel=< ch >&code=AutoPanOff&arg1=0&arg2=0&arg3=0
Method	GET
Description	Stop pan.
Example	http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=AutoPanOff&arg1=0&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ch is PTZ channel index which starts from 1. Range is [1 — n].

8.1.11 [Config] PTZ Auto Movement

- Get PTZ auto movement configuration

Table 8-22

Syntax	<code>http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=PtzAutoMovement</code>
Method	GET
Description	Get PTZ auto movement configuration.
Example	<code>http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=PtzAutoMovement</code>
Success Return	<pre>table.PtzAutoMovement[port][Task].Enable = true table.PtzAutoMovement[port][Task].TimeSection[week][section] = 0 00:00:00-23:59:59 ... table.PtzAutoMovement[port][Task].Function = None table.PtzAutoMovement[port][Task].ScanId = 0 table.PtzAutoMovement[port][Task].PresetId = 0 table.PtzAutoMovement[port][Task].PatternId = 0 table.PtzAutoMovement[port][Task].TourId = 0 table.PtzAutoMovement[port][Task].AutoHoming.Time = 30 table.PtzAutoMovement[port][Task].SnapshotEnable = false table.PtzAutoMovement[port][Task].SnapshotDelayTime = 30</pre>
Comment	<p>Parameters in Response:</p> <p>port is PTZ port index, start from 0.</p> <p>Task is the number of task, start from 0.</p> <p>week: from 1 to 7.</p> <p>section: time section, from 0 to 5.</p>

- Set PTZ auto movement

Table 8-23

Syntax	<code>http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue> [&<paramName>=<paramValue> ...]</code>
Method	GET
Description	Set PTZ auto movement.
Example	<code>http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&PtzAutoMovement[0][0].Function=Preset&PtzAutoMovement[0][0].PresetId=1</code>
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>In table below,</p> <p>head=PtzAutoMovement[port][task]</p> <p>port is PTZ port index, start from 0.</p> <p>task is the number of task, start from 0.</p> <p>week: from 1 to 7.</p> <p>section: time section, from 0 to 5.</p>

Appendix:

ParamName	ParamValue type	Description
<code>head.Enable</code>	bool	Enable/Disable PtzAutoMovement
<code>head.TimeSection</code>	timeSchedule	timeSchedule[week][section]=1 10:00:00-11:00:00 ...
<code>head.Function</code>	string	Range is {Scan, Preset, Pattern, Tour}.
<code>head.ScanId</code>	integer	Scan Id, start from 1
<code>head.PresetId</code>	integer	Preset Id, start from 1
<code>head.PatternId</code>	integer	Pattern Id, start from 1
<code>head.TourId</code>	integer	Tour Id, start from 1
<code>head.AutoHoming.Time</code>	integer	Recover time, unit is second.
<code>head.SnapshotEnable</code>	bool	Enable/Disable Snap, when "Fuction" is "Preset".
<code>head.SnapshotDelayTime</code>	integer	Delay time of snap, when "Fuction" is "Preset".

8.1.12 PTZ Restart

Table 8-24

Syntax	http://<server>/cgi-bin/ptz.cgi?action=start&channel=< ch >&code=Restart&arg1=0&arg2=0&arg3=0
Method	GET
Description	PTZ restart.
Example	http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=Restart&arg1=0&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ch is PTZ channel index which starts from 1. Range is [1 — n].

8.1.13 PTZ Reset

Table 8-25

Syntax	http://<server>/cgi-bin/ptz.cgi?action=start&channel=< ch >&code=Reset&arg1=0&arg2=0&arg3=0
Method	GET
Description	PTZ reset.
Example	http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=Reset&arg1=0&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ch is PTZ channel index which starts from 1. Range is [1 — n].

8.1.14 OSD Menu

- Enter the menu

Table 8-26

Syntax	<code>http://<server>/cgi-bin/ptz.cgi?action=start&channel=<ch>&code=Menu&arg1=0&arg2=0&arg3=0</code>
Method	GET
Description	Enter the menu.
Example	<code>http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=Menu&arg1=0&arg2=0&arg3=0</code>
Success Return	OK
Comment	Parameters in URL: ch is PTZ channel index which starts from 1. Range is [1 — n].

- Exit the menu

Table 8-27

Syntax	<code>http://<server>/cgi-bin/ptz.cgi?action=start&channel=<ch>&code=Exit&arg1=0&arg2=0&arg3=0</code>
Method	GET
Description	Exit the menu.
Example	<code>http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=Exit&arg1=0&arg2=0&arg3=0</code>
Success Return	OK
Comment	Parameters in URL: ch is PTZ channel index which starts from 1. Range is [1 — n].

- Confirm

Table 8-28

Syntax	<code>http://<server>/cgi-bin/ptz.cgi?action=start&channel=<ch>&code=Enter&arg1=0&arg2=0&arg3=0</code>
Method	GET
Description	Confirm the menu.
Example	<code>http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=Enter&arg1=0&arg2=0&arg3=0</code>
Success Return	OK
Comment	Parameters in URL: ch is PTZ channel index which starts from 1. Range is [1 — n].

- Start the basic operation of menu

Table 8-29

Syntax	http://<server>/cgi-bin/ptz.cgi?action=start&channel=< ch >&code=< code >&arg1=< arg1 >&arg2=< arg2 >&arg3=< arg3 >
Method	GET
Description	Start the basic operation of menu.
Example	http://192.168.1.250/cgi-bin/ptz.cgi?action=start&channel=1&code=MenuUp&arg1=0&arg2=0&arg3=0
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>ch is PTZ channel index which starts from 1. Range is [1 — n].</p> <p>code is PTZ operation, and arg1, arg2, arg3 are arguments of operation.</p> <p>code and argN values are listed in table below.</p>

Appendix:

Code	Code description	arg1	arg2	arg3
MenuUp	MenuUp	0	0	0
MenuDown	MenuDown	0	0	0
MenuLeft	MenuLeft	0	0	0
MenuRight	MenuRight	0	0	0

8.1.15 [Config] Set up electronic PTZ

Request URL	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]			
Method	GET			
Request Params (key=value format at URL)				
Name	Type	R/O	Description	Example
EptzLink[channel].Enable	bool	O	Whether to enable the electronic PTZ function in this channel. Channel indicates the channel number	true
EptzLink[channel].DisplayMode	char[32]	O	Display mode, used to select the number of channels to track; Options are: "original" - normal mode; "Oneplusone" - 1 + 1 mode; "Oneplusthree" - 1 + 3 mode; "Oneplusfive" - 1 + 5 mode	"Original"
EptzLink[channel].TrackEnable	bool	O	Whether to start linkage tracking; True - on, False close	true
EptzLink[channel].Tra	char[32]	O	Tracking duration options,	"Manual"

ckTime			including full-time tracking and manual configuration of tracking duration. "Fulltimetrack": full-time tracking, indicating that the tracking duration lasts until the target disappears; "Manual": customize the tracking duration. If this mode is selected, it will be tracked according to the user configured tracking duration.	
EptzLink[channel].ManualTrackTime[i]	uint16	O	If tracktime is in "manual" mode, it will be tracked according to the tracking duration range configured by the user in this field ([minimum tracking time, maximum tracking time]), which can be configured for 5-300 seconds. The maximum duration of tracking is the disappearance time of the target. For example, if the tracking duration is set to 50 seconds and the target disappears in 30 seconds, the tracking will stop). 'I' represents the index of the array, and 'i' can take 0 or 1	5
EptzLink[channel].TrackRect[i][j][k]	uint32	O	The coordinate information of the tracking target frame represents one of the points. The first dimension represents the display mode, the second dimension represents the number of frames, and the third dimension is the bounding box of the rectangular frame, which is normalized to the 8192 coordinate system.	2047

Example

[http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&EptzLink\[0\].Enable=true](http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&EptzLink[0].Enable=true)

8.1.16 Get View Range Status

Request URL	http://<server>/cgi-bin/api/ptz/getViewRangeStatus			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Channel	int32	R	Video channel number, starting from 0	0
Request Example				
{ "Channel": 0 }				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
status	object	否	view range status	
+Distance	double	否	Visual distance in meters	20.0
+AngelH	double	否	Horizontal viewing angle Unit: degrees	30.5
+AzimuthH	double	否	Horizontal azimuth angle (The angle between the central axis of the included angle projected by the visual field on the horizontal plane and the reference axis on the horizontal plane) Expressed in radians, normalized to - 1~1. Turn counterclockwise for positive direction	0.5
+AngelV	double	否	Vertical viewing angle Unit: degrees	30.5
+AzimuthV	double	否	Vertical azimuth angle (The angle between the central axis of the included angle projected by the visual field on the vertical plane and the angle of the reference axis on the vertical plane)	0.5

			Expressed in radians, normalized to - 1~1.	
+InclinationH	double	否	Horizontal inclination angle (The included angle between the equipment horizontal datum plane and the horizontal plane) Unit: degrees Normalized to - 1~1.	0.5

Response Example

```
{
  "status": {
    "Distance": 20.0,
    "AngelH": 30.5,
    "AzimuthH": 0.5,
    "AngelV": 30.5,
    "AzimuthV": 0.5,
    "InclinationH": 0.5
  }
}
```

8.2 Wiper

8.2.1 Move Continuously

Make the wiper continuously move until you call the stopMove function to stop it.

Request URL	http://<server>/cgi-bin/rainBrush.cgi?action=moveContinuously			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	Video channel number, starting from 1; the default value is 1	1
interval	int	R	Interval for the motion of the wiper; unit: second	5
Request Example				
http://192.168.1.250/cgi-bin/rainBrush.cgi?action=moveContinuously&channel=1&interval=5				

Response Params (OK in body)

Response Example

OK

8.2.2 Stop Move

Stop the motion of the wiper.

Request URL	http://<server>/cgi-bin/rainBrush.cgi?action=stopMove			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	Video channel number, starting from 1; the default value is 1	1
Request Example				
http://192.168.1.250/cgi-bin/rainBrush.cgi?action=stopMove&channel=1				

Response Params (OK in body)
Response Example
OK

8.2.3 Move Once

Control the wiper to move once.

Request URL	http://<server>/cgi-bin/rainBrush.cgi?action=moveOnce			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	video channel index which starts from 1, default 1 if not specified.	1
Request Example				
http://192.168.1.250/cgi-bin/rainBrush.cgi?action=moveOnce&channel=1				

Response Params (OK in body)
Response Example
OK

8.3 Illuminator

The following commands are applicable to non-intelligent illuminators.

8.3.1 [Config] Visible-light Illuminator

Template	http://<server>/cgi-bin/configManager.cgi?action=setConfig&SignLight[0].onCycle=30			
Method	GET			
Parameter Format	key=value format in URL			
Parameter	Type	Required	Description	Example

onCycle	Uint32	Yes	Range[0-100]	30
Example				
http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&SignLight[0].onCycle=30				

Parameter Format	OK at body			
Parameter	Type	Required	Description	Example
Example				
OK				

8.4 Flashlight

8.4.1 [Config] Flashlight Config

Configuration parameters of the flashing light (Not recommended).

Config Data Params				
Name	Type	R/O	Description	Example
FlashLight	object	R	Configuration parameters of the flashing light	
+Enable	bool	R	Whether to enable the flashing light	true
+Brightness	int	R	Brightness; range [0, 100]	50
+TimeSection	char[7][24][32]	R	Two-dimensional array in strings. The first dimension of the array is the day of the week; range: [0-6] (Sunday–Saturday). The second dimension is the period index. One day is divided into multiple periods, and the range is [0–23]. Each period is set as a string in the format of mask hh: mm: ss-hh: mm: ss Mask: Value range: [0, 1] hh: Hour; range: [0–24] mm: Minute; range: [0–59] ss: Second; range: [0–59] Mask value: 0: Disable the parameter in the period 1: Enable the parameter in the period For example: TimeSection[1][0]=1 12:00:00—18:00:00 Meaning the flashing light works from 12:00:00 through 18:00:00 on Monday.	[["1 00:00:00-23:59:59", "0 0:00:00-23:59:59", "0 0:00:00-023:59:59", "0 0:00:00-23:59:59", "0 0:00:00-23:59:59", "0 0:00:00-23:59:59", "0 0:00:00-23:59:59"]]

For getting and setting the configurations, see 4.2.1 Get and Set Configure”.

Get Config Request Example
http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=FlashLight
Get Config Response Example

```

table.FlashLight.Enable=true
table.FlashLight.Brightness=50
table.FlashLight.TimeSection[0][0]=1 00:00:00-23:59:59
table.FlashLight.TimeSection[0][1]=0 00:00:00-23:59:59
...
table.FlashLight.TimeSection[6][5]=0 00:00:00-23:59:59
...

```

Set Config Request Example

http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&FlashLight.Enable=true&FlashLight.Brightness=50&FlashLight.TimeSection[1][0]=1%2012:00:00-18:00:00

Set Config Response Example

OK

8.5 Coaxial Control IO

8.5.1 Control White Light or Speaker

Send commands for controlling the white light and speaker

Request URL	http://<server>/cgi-bin/coaxialControlIO.cgi?action=control		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	It is the video channel number for the white light, and the audio output channel for the speaker. The default value is 0.
info	object[]	R	Operation details
+Type	int	R	Operation type 1: White light 2: Speaker
+IO	int	R	Switch 1: On 2: Off
+TriggerMode	int	R	Trigger Mode 1: Linked trigger 2: Manual trigger
Request Example			
http://192.168.1.250/cgi-bin/coaxialControlIO.cgi?action=control&channel=1&info[0].Type=1&info[0].IO=1&info[0].TriggerMode=2			

Response Params (OK in body)

Response Example

OK

8.5.2 Get White Light and Speaker Status

Get the status of the white light and the speaker.

Request URL	http://<server>/cgi-bin/coaxialControlIO.cgi?action=getstatus			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	The video channel number which starts from 1, and the default value is 1.	1
Request Example				
http://192.168.1.250/cgi-bin/coaxialControlIO.cgi?action=getstatus&channel=1				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
status	object	R	Return status information	
+whitelight	char[4]	R	White light status, "on" or "off"	"on"
+speaker	char[4]	R	Speaker status, "on" or "off"	"on"
Response Example				
status.whitelight=on status.speaker=on				

8.6 Pir Alarm

8.6.1 [Config] Pir Parameter

- Get pir parameter

Table 8-30

Syntax	http://<server>/cgi-bin/pirAlarm.cgi?action=getPirParam&[channel=<channelNo>]
Method	GET
Description	Get pir parameter.
Example	http://192.168.1.250/cgi-bin/pirAlarm.cgi?action=getPirParam&channel=1
Success Return	head.Enable=true head.PirLink.LightingLink.Enable=true head.PirLink.LightingLink.LightLinkType=Filcker head.PirLink.LightingLink.FilckerIntevalTime=5 head.PirLink.LightingLink.LightDuration=10 head.PirLink.LightingLink.WhiteLightTimeSection=TimeSection head.PirLink.TimeSection[weekday][0]=1 00:00:00-24:00:00 head.PirLink.TimeSection[weekday][1]=0 02:00:00-24:00:00 head.PirLink.TimeSection[weekday][2]=0 03:00:00-24:00:00 head.PirLink.TimeSection[weekday][3]=0 04:00:00-24:00:00 head.PirLink.TimeSection[weekday][4]=0 05:00:00-24:00:00 head.PirLink.TimeSection[weekday][5]=0 06:00:00-24:00:00

	<pre> head.RecordEnable=true head.RecordChannels=[0, 1, 2] head.RecordLatch=10 head.AlarmOutEnable=true head.AlarmOutChannels=[1, 4] head.AlarmOutLatch=10 head.SnapshotEnable=true head.SnapshotChannels=[2, 4] head.MailEnable=true head.AlarmBellEnable=true head.AlarmBellLatch=10 head.Dejitter=0 head.LogEnable=true head.DetectWindow[0].Level=3 head.DetectWindow[0].Id=0 head.DetectWindow[0].Name=Region0 head.DetectWindow[0].Sensitive=58 head.DetectWindow[0].Threshold=4 head.DetectWindow[0].Region[0]=3932160 head.DetectWindow[0].Region[1]=3932160 head. DetectWindow [1]... </pre>
Comment	<p>if a device has one or more pir, PirParam is a array</p> <p>channel : video channel index</p> <p>weekday : range is [0-6] (Sunday - Saturday)</p> <p>head=configEx[channelNo]</p>

- set pir parameter

Table 8-31

Syntax	http://<server>/cgi-bin/pirAlarm.cgi?action=setPirParam&[channel=< channelNo >]&< paramName >=< paramValue >...&< paramName >=< paramValue >
Method	GET
Description	Set pir parameter.
Example	http://192.168.1.250/cgi-bin/pirAlarm.cgi?action=setPirParam&channel=1&configEx[1].Enable=true&configEx[1].PirLink.LightingLink.Enable=true&...
Success Return	OK
Comment	<p>channel : video channel index</p> <p>head=configEx[channelNo]</p>

Appendix:

ParamName	ParamValue type	Description
head .Enable	bool	Enable/Disable motion detect feature in a channel.

ParamName	ParamValue type	Description
head.DetectWindow[WinNum].Level	integer	<p><i>WinNum</i></p> <p>Index of detect window, there are 4 detect windows at present. Each window is divided into 18 lines and 22 blocks/line.</p> <p>Range is [1—6].</p> <p>Sensitivity of pir</p> <p>1: lowest sensitivity.</p> <p>6: highest sensitivity.</p>
head.DetectWindow[WinNum].Id	integer	It is the Id of a detect window.
head.DetectWindow[WinNum].Name	string	It is the name of a detect window.
head.DetectWindow[WinNum].Sensitive	integer	<p>Range is [0—100].</p> <p>It presents more sensitive if the value is larger.</p>
head.DetectWindow[WinNum].Threshold	integer	<p>Range is [0—100].</p> <p>It presents the threshold value when trigger motion detect.</p>
head.DetectWindow[WinNum].Region[LineNum]	integer	<p><i>LineNum</i></p> <p>Index of region, region is divided into lines and each line has several blocks, a line is described by a 32 bit integer, a bit for a block.</p> <p>0=Line 1 1=Line 2</p> <p>Currently, region is divided into 18 lines and 22 blocks/line.</p> <p>A bit describes a block in the line.</p> <p>Bit = 1: motion in this block is monitored.</p> <p>Example:</p> <p>MotionDetect[0].Region[0] = 4194303 (0x3FFFFF):: motion in channel 0 line 0's 22 blocks is monitored.</p> <p>MotionDetect[0].Region[1] =0: motion in line 1's 22 blocks is not monitored.</p> <p>MotionDetect[0].Region[17] = 3: in the last line of channel 0, motion in the left two blocks is monitored.</p>

ParamName	ParamValue type	Description
head.TimeSection[wd][ts]	string	<p>wd (week day) range is [0—6] (Sunday - Staurdy) ts (time section) range is [0 — 23], timesection table index.</p> <p>Format: mask hh:mm:ss-hh:mm:ss Mask: [0—65535], hh: [0—24], mm: [0—59], ss: [0—59] Mask indicates record type by bits: Bit0: regular record Bit1: motion detection record Bit2: alarm record Bit3: card record</p>
head.PirLink.RecordChannels[ch]	Integer	<p>Range is {0, 1} 0 – do not record on video channel ch 1 – record on video channel ch</p>
head.PirLink.RecordEnable	bool	Enable/Disable record function.
head.PirLink.RecordLatch	integer	<p>Range is [10—300]. Unit is seconds, indicates the time to record after input alarm is cleared.</p>
head.PirLink.AlarmOutChannels[ch]	integer	<p>Range is {0, 1}, ch is alarm out channel index. 0 — do not output alarm at alarm out channel ch 1 — output alarm at alarm out channel ch</p>
head.PirLink.AlarmOutEnable	bool	Enable/Disable alarm out function.
head.PirLink.AlarmOutLatch	Integer	<p>Range is [10—300]. Unit is seconds, indicates the time to output alarm after input alarm is cleared.</p>
head.PirLink.SnapshotChannels[ch]	integer	<p>Range is {0, 1} 0 — do not snapshot on video channel ch 1 — snapshot on video channel ch</p>
head.PirLink.SnapshotEnable	bool	Enable/Disable snapshot function.
head.PirLink.Dejitter	integer	<p>Range is [0—255]. Alarm signal dejitter seconds. Alarm signal change during this period is ignored.</p>
head.PirLink.MailEnable	bool	Enable/Disable mail send for alarm.
head.PirLink.AlarmBellEnable	bool	Enable/Disable mail send for alarm.
head.PirLink.AlarmBellLatch	integer	<p>Range is [10, 300] Unit is seconds</p>
head.PirLink.LogEnable	bool	Enable/Disable log for alarm.

8.7 SCADA

8.7.1 Searching for SCADA Attributes

Request URL	http://<server>/cgi-bin/api/SCADA/getAttribute			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
condition	object	R		
+DeviceId	char[32]	R	Device ID	"01001010111"
+ID	char[]	No	An array of corresponding monitoring point ID. If there is no node, return all IDs under DeviceId.	["01001010111", "01001010112",...]
Request Example				
{				
"condition": {				
"DeviceId": "01001010111",				
"ID": ["01001010111", "01001010112",...]				
}				
}				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
info	object[]	R		
+SignalName	char[128]	R	Point name	"Environment temperature"
+Unit	char[20]	R	Unit	"°C"
+Type	char[20]	R	Type of signal point YC: remote metering; analog input. YX: remote signaling; digital input. YT: remote regulation; analog output. YK: remote control; digital output.	"YC"
+ID	char[32]	R	Signal point ID	"01001010111"
+StartDelay	uint	O	Start delay of alarms, unit: second	120
+StopDelay	uint	O	End delay of alarms, unit: second	20
+Period	uint	O	Storing and reporting period,	180

			unit: second	
+Threshold	float	O	Alarm threshold	40.0
+AlarmWaveVal	float	O	Alarm hysteresis. Only when the hysteresis range is exceeded, the alarm can be restored. Alarm recovery and alarm delay do not take effect at the same time.	2
+AbsoluteVal	float	O	Absolute threshold. Stores and reports continuous data points (AI, AO).	0.2
+RelativeVal	float	O	Percentage threshold. Stores and reports continuous data points (AI, AO). It is used when the absolute threshold is disabled or is "0".	0
+Status	uint	R	Data status of signal point 0: normal 1: Level-1 alarm 2: Level-2 alarm 3: Level-3 alarm 4: Level-4 alarm 5: Operation event 6: Invalid data	0
+DisplayOptions	uint	O	Displays data: BIT0: Displays the switch. 0: No; 1: Yes. BIT1: Displays the switch when scrolling. 0: No; 1: Yes.	0x00000000
+Valid	bool	R	Invalid signal point true: Yes false: No	true
+Delay	uint32	O	Alarm delay time. When an alarm is triggered, the device response such as reporting and linkage is delayed. Unit: Second.	120
+Describe	char[]	O	Point description. For example, 0&Normal; 1&Alarm. Supports up to 120 characters.	"Environment temperature"
+IECCode	char[]	O	Power 104 protocol point number. Supports up to 16 characters.	"12345"
+HJCode	char[]	O	Environment 212 protocol point number. Supports up to 16 characters.	"a01001"
Response Example				

```
{
  "info": [
    {
      "SignalName": "Environment temperature",
      "Unit": "°C",
      "Type": "YC",
      "ID": "01001010111",
      "StartDelay": 120,
      "StopDelay": 20,
      "Period": 180,
      "Threshold": 40.0,
      "AlarmWaveVal": 2,
      "AbsoluteVal": 0.2,
      "RelativeVal": 0,
      "Status": 0,
      "DisplayOptions": 0x00000000,
      "Valid": true,
      "Delay": 120,
      "Describe": "Environment temperature",
      "IECCode": "12345",
      "HJCode": "a01001"
    }, ...
  ]
}
```

8.7.2 Configuring SCADA Attributes

You can use this method to disable or enable the point of SCADA devices.

Request URL	http://<server>/cgi-bin/api/SCADA/setAttribute			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
DeviceId	char[32]	R	Detector	"01001010111"
points	object[]	R		
+ID	char[32]	R		"01001010110"
+SignalName	char[128]	O	Point name	"Environment temperature"
+StartDelay	uint	O	Start delay of alarms, unit: second	120
+StopDelay	uint	O	End delay of alarms, unit: second	20
+Period	uint	O	Storing and reporting period, unit: second	180
+Threshold	float	O	Alarm threshold	40.0
+AlarmWaveVal	float	O	Alarm hysteresis. Only when the hysteresis range is exceeded, the alarm can be restored. Alarm recovery and	2

			alarm delay do not take effect at the same .	
+AbsoluteVal	float	O	Absolute threshold. Stores and reports continuous data points (AI, AO).	0.2
+RelativeVal	float	O	Percentage threshold. Stores and reports continuous data points (AI, AO). It is used when the absolute threshold is disabled or is "0".	0
+Status	enumint	O	Data status of signal point enumint{ 0: normal 1: Level-1 alarm 2: Level-2 alarm 3: Level-3 alarm 4: Level-4 alarm 5: Operation event 6: Invalid data }	0
+DisplayOptions	uint	O	Displays data: BIT0: Displays the switch or not. 0: No; 1: Yes. BIT1: Displays the switch when scrolling. 0: No; 1: Yes.	0x00000000
+Valid	bool	R	Valid signal point. true: Valid. false: Invalid.	true

Request Example

```
{
  "DeviceId": "01001010111",
  "points": [
    {
      "ID": "01001010110",
      "SignalName": "Environment temperature",
      "StartDelay": 120,
      "StopDelay": 20,
      "Period": 180,
      "Threshold": 40.0,
      "AlarmWaveVal": 2,
      "AbsoluteVal": 0.2,
      "RelativeVal": 0,
      "Status": 0,
      "DisplayOptions": 0x00000000,
      "Valid": true
    }, ...
  ]
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
state	object	O		
+Success	char[][]	O	The list with successful IDs	["01001010111"]
+Fail	char[][]	O	The list with failed IDs	["01001010122"]
Response Example				
{				
"state": {				
"Success": ["01001010111"],				
"Fail": ["01001010122"]				
}				
}				

8.7.3 Obtaining Real-time Data of Monitoring Points

Request URL	http://<server>/cgi-bin/api/SCADA/get			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
condition	object	R		
+DeviceId	char[16]	R	Detector ID	"01001010111"
+ID	char[][]	O	An array of corresponding monitoring point ID If there is no node, return all IDs under the DeviceId.	["01001010111", "01001010112",...]]
+IsHandle	bool	O	The returned data is processed (such as filtering invalid data). "false": No, "true": Yes. Default: false.	false
Request Example				
{				
"condition": {				
"DeviceId": "01001010111",				
"ID": ["01001010111", "01001010112",...],				
"IsHandle": false				
}				
}				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
info	object[]	R		
+Type	enumchar[32]	O	Point type enumchar[32]{ YC: Remote metering; analog input. YX: Remote signaling; digital	"YC"

			<p>input.</p> <p>YT: Remote regulation; analog output.</p> <p>YK: Remote control; digital output.</p> <p>}</p>	
+ID	char[32]	R	Point	"01001010111"
+PointName	char[32]	O	Matches the point table name.	"aaa"
+MeasuredVal	float	R	<p>The actual measured value.</p> <p>You can change the type.</p> <p>YX: Integer</p> <p>YC: Floating-point number</p>	213.1
+SetupVal	float	O	<p>The configured value. You can change the type.</p> <p>YK: Integer</p> <p>YT: Floating-point number</p>	123.0
+Status	enumint	O	<p>Data status</p> <p>enumint{</p> <p>0: Normal</p> <p>1: Level-1 alarm</p> <p>2: Level-2 alarm</p> <p>3: Level-3 alarm</p> <p>4: Level-4 alarm</p> <p>5: Operation Event</p> <p>6: Invalid data</p> <p>}</p>	0
+RecordTime	char[20]	O	Collection time	"2015-1-3 10:10:45"

Response Example

```
{
  "info": [
    {
      "Type": "YC",
      "ID": "01001010111",
      "PointName": "aaa",
      "MeasuredVal": 213.1,
      "SetupVal": 123.0,
      "Status": 0,
      "RecordTime": "2015-1-3 10:10:45"
    }, ...
  ]
}
```

8.7.4 Configuring Monitoring Points

Request URL	http://<server>/cgi-bin/api/SCADA/set		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description

DeviceId	char[32]	R	Detector	"01001010111"
points	object[]	R		
+Type	char[32]	O	Point type YT YT	"YT"
+ID	char[64]	R	Point number	"01001010111"
+SetupVal	float	R	The configured value YT: Floating-point number YK: Integer	12.3

Request Example

```
{
    "DeviceId": "01001010111",
    "points": [
        {
            "Type": "YT",
            "ID": "01001010111",
            "SetupVal": 12.3
        }, ...
    ]
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
state	object	O		
+Success	char[][]	O	The list with successfully controlled and adjusted IDs	["01001010111"]
+Fail	char[][]	O	The list with unsuccessfully controlled and adjusted IDs	["01001010122"]

Response Example

```
{
    "state": {
        "Success": ["01001010111"],
        "Fail": ["01001010122"]
    }
}
```

8.7.5 Starting Searching for Historical Data

Start searching for historical data based on device ID and time, and get search token.

Request URL	http://<server>/cgi-bin/api/SCADA/startFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
condition	object	R	Search conditions	
+StartTime	char[20]	R	Start time	"2015-1-2 10:10:45"
+EndTime	char[20]	O	End time If you don't enter content is this field, it means up to now.	"2015-1-3 10:10:45"
+DeviceId	char[32]	R	Device ID	"01001010111"

+ID	char[32]	O	ID of corresponding monitoring point	"01001010111"
Request Example				
{				
"condition": {				
"StartTime": "2015-1-2 10:10:45",				
"EndTime": "2015-1-3 10:10:45",				
"DeviceId": "01001010111",				
"ID": "01001010111"				
}				
}				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
token	int	R	Obtained search token	46878
totalCount	int	O	Total number of qualified results	3333
Response Example				
{				
"token": 46878,				
"totalCount": 3333				
}				

8.7.6 Obtaining Historical Data

Obtain historical data based on search token.

Request URL	http://<server>/cgi-bin/api/SCADA/doFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
token	int	R	Search token	46878
beginNumber	int	R	Start number of the search. The search starts from the "beginNumber" records, and returns predefined number of records. 0<=beginNumber<=totalCount-1	0
count	int	R	Number of traffic statistics for each search	24
Request Example				
{				
"token": 46878,				
"beginNumber": 0,				
"count": 24				
}				

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
found	int	R	Number of searched entries	12
info	object[]	R	Traffic statistics It is an array and each element represents a traffic record that satisfies the condition.	
+Type	char[32]	R	Point type enumchar[32]{ YC: remote metering; analog input YX: Remote signaling; digital input: YT: Remote regulation; analog output YK: Remote control; digital output }	"YC"
+ID	char[32]	R	Monitoring point ID	"01001010111"
+MeasuredVal	float	R	Value You can change this filed type, which is relevant to Type filed. When Type is YC, this value is floating point number; When Type is YX, this value is integer.	213.1
+SetupVal	float	R	Configured value You can change this filed type, which is relevant to Type filed. When Type is YC, this value is floating point number; When Type is YX, this value is integer.	123.0
+Status	int	R	Point status	0
+RecordTime	char[20]	R	Record time	"2015-1-3 10:10:45"
Response Example				
<pre>{ "found": 12, "info": [{ "Type": "YC", "ID": "01001010111", "MeasuredVal": 213.1, "SetupVal": 123.0, "Status": 0, "RecordTime": "2015-1-3 10:10:45" }, ...] }</pre>				

8.7.7 Stopping Searching for Historical Data

Request URL	http://<server>/cgi-bin/api/SCADA/stopFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
token	int	R	Search token	46878
Request Example				
{ "token": 46878 }				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				
{}				

8.7.8 Obtaining IDs of External Devices Connected to the Host

Request URL	http://<server>/cgi-bin/api/SCADA/getDeviceList			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Request Example				
{}				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
devices	object[]	R	Device ID array	
+DeviceId	char[32]	R	Device ID	"0101012345"
+DevName	char[32]	O	Device name	"UPS1"
+DevCode	char[8]	O	Device model	"1801"
+Slot	int	O	Virtual slot number In general, 0 means analog channel and 1 means digital channel. If it is greater than 1, it means RS-485, RS-232, or network channel. The actual channel number = virtual slot number – 2.	0
+Level	int	O	The detector address configured by the environment surveillance server. A slot lot can be connected to multiple detectors, and different	1

		detectors are distinguished by address.	
Response Example			
{	"devices": [{	"Deviceld": "0101012345",	

```

        "DevName": "UPS1",
        "DevCode": "1801",
        "Slot": 0,
        "Level": 1
    },...{}]
}
```

8.8 Gyro

8.8.1 get gyroscope info

Request URL	http://<server>/cgi-bin/api/Gyro/getData			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Request Example				
{}				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
detail	object	R		
+AccelX	double	R	Gravitational acceleration in X direction, unit m/s2	15,3
+AccelY	double	R	Gravitational acceleration in Y direction, unit m/s2	20.7
+AccelZ	double	R	Gravitational acceleration in Z direction, unit m/s2	10.5
+Pitch	double	O	Pitch angle, in degrees range[-90, 90]	1.3
+Roll	double	O	roll angle, in degrees range[-180, 180]	23.6
+Yaw	double	O	yaw angle, in degrees range[-180, 180]	120.2
+AngularSpeedX	double	R	X-direction angular velocity (rad/s)	2.5
+AngularSpeedY	double	R	Y-direction angular velocity (rad/s)	3.3
+AngularSpeedZ	double	R	Z-direction angular velocity (rad/s)	3.6
Response Example				
{				

```
"detail": {  
    "AccelX": 15.3,  
    "AccelY": 20.7,  
    "AccelZ": 10.5,  
    "Pitch": 1.3,  
    "Roll": 23.6,  
    "Yaw": 120.2,  
    "AngularSpeedX": 2.5,  
    "AngularSpeedY": 3.3,  
    "AngularSpeedZ": 3.6  
}  
}
```

9. Video Analyse API

9.1 Video Analyse Event

9.1.1 [Event] LeftDetection

Table 9-1

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When detect some object left, send this event.		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
Object	object	R	The object that left.
+BoundingBox	Array<int>	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.
[Example]			
Event	<pre>--<boundary> Content-Type: text/plain Content-Length: </length> Code=LeftDetection;action=pulse;index=0;data={ "Object": { "BoundingBox": [2992,1136,4960,5192] } } --<boundary></pre>		

9.1.2 [Event] TakenAwayDetection

Table 9-2

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When detect some object was taken away, send this event.		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
Object	object	R	The object that was taken away.
+BoundingBox	Array<int>	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.
[Example]			
Event	<pre>--<boundary></pre>		

	Content-Type: text/plain Content-Length: <length> Code=TakenAwayDetection;action=pulse;index=0;data={ "Object": { "BoundingBox": [2992,1136,4960,5192] } } --<boundary>
--	--

9.1.3 [Event] WanderDetection

Table 9-3

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When detect some object was wandering, send this event		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
Objects	Array<object>	R	The objects that was wandering.
+BoundingBox	Array<int>	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.
Tracks	Array<Array<Array<int>>>	O	The object wandering tracks, array of polyline, one polyline for one object, polyline is array of points, point is array of two int, x's value and y's value. Coordinate remap to 0 — 8192.
DetectRegion	Array<Array<int>>	R	The detection region, the first array is point list, max item is 20, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.

[Example]	--<boundary> Content-Type: text/plain Content-Length: <length> Code=WanderDetection;action=start;index=0;data={ "Objects": [{ "BoundingBox": [2992,1136,4960,5192] }, { "BoundingBox": [4392,4136,6960,6512] }, { ... }, ...], "Tracks": [[[1202, 576], [1456, 863], [1921, 1204], [2341, 1823], [3512, 2314]], [[2214, 3412], [3153, 3674], [4512, 4213]], [...], ...], "DetectRegion": [[1292,3469], [6535,3373], ...] } --<boundary>
Event	

9.1.4 [Event] StayDetection

Table 9-4

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When detect some object was stay, send this event		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
Object	object	O	The object that was stay.
+BoundingBox	Array<int>	O	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.
Objects	array<object>	O	If detect several object, store in this array.
+BoundingBox	Array<int>	O	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.
DetectRegion	Array<Array<int>>	O	The detection region, the first array is point list, max item is 20, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.
AreaID	int	O	The area id, begin from 1, if omit, means single area.
PresetID	int	O	The preset id, if omit, means preset is unknown.

[Example]

Event	--<boundary> Content-Type: text/plain Content-Length: <length> Code=StayDetection;action=start;index=0;data={ "Object": { "BoundingBox": [2992,1136,4960,5192] }, "Objects": [{ "BoundingBox": [2992,1136,4960,5192] }, { "BoundingBox": [4392,4136,6960,6512] }, {...}, ...], "DetectRegion": [[1292,3469], [6535,3373], ...], "AreaID" : 2 } --<boundary>
-------	--

9.1.5 [Event] HumanTrait

Table 9-5

Usage	Refer to "4.4.3 Subscribe to Snapshot "for how to subscribe event		
Description	When detect a human trait, send this event.		
[Event Params] (key=value format)			
Name	Type	R/O	Param Description
Events	array<object>	O	Event info array.
+EventBaseInfo	object	R	Base info of event.
++Code	String	R	Event Code. It should be HumanTrait .
++Action	String	R	Event Action. It can be: "Start", "Stop", "Pulse".

++Index	int	O	The channel index relate to this event, start from 0.
+HumanAttributes	object	O	The human attributes.
++BoundingBox	Array<int>	O	The detected human bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
++Sex	string	O	Sex, can be "Man", "Woman", "Unknown".
++Age	int	O	Age.
++Angle	int	O	Angle, 0: unknown, 1: front, 2: side, 3: back.
++CoatColor	string	O	Coat color, can be: "White", "Orange", "Pink", "Black", "Red", "Yellow", "Gray", "Blue", "Green", "Purple", "Brown", "Sliver", "Darkviolet", "Maroon", "Dimgray", "Whitesmoke", "Darkorange", "Mistyrose", "Tomato", "Olive", "Gold", "Darkolivegreen", "Chartreuse", "Greencyellow", "Forestgreen", "Seagreen", "Chartreuse", "Deepskyblue", "Cyan", "Other".
++CoatType	int	O	Coat type, 0: unknown, 1: long sleeve, 2: short sleeve.
++TrousersColor	string	O	Trousers color, value can be that of CoatColor.
++TrousersType	int	O	Trousers type, 0: unknown, 1: long pants, 2: short pants, 3: skirt.
++HasHat	int	O	Has hat or not, 0: unknown, 1: not has hat, 2: has hat.
++HasBag	int	O	Has bag or not, 0: unknown, 1: not has bag, 2: has bag.
++HasUmbrella	int	O	Has umbrella or not, 0: unknown, 1: not has umbrella, 2: has umbrella.
++Bag	int	O	Bag type, 0: unknown, 1: handbag, 2: shoulder bag, 3: knapsack, 4: draw-bar box.
++UpperPattern	int	O	Upper clothes pattern, 0: unknown, 1: pure color, 2: stripe, 3: pattern, 4: gap, 5: grid.
++HairStyle	int	O	Hair style, 0: unknown, 1: long hair, 2: short hair, 3: ponytail, 4: updo, 5: hiddened.
++Cap	int	O	Cap style, 0: unknown, 1: normal cap, 2: helmet.
+FaceAttributes	object	O	If the human's face can be detected, find it's attributes.
++BoundingBox	Array<int>	O	The detected face bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
++Sex	string	O	Sex, it can be "Man", "Woman", "Unknown".
++Age	int	O	Age.
++Feature	array<string>	O	Face feature, can be some of the following : "WearGlasses", "SunGlasses", "NoGlasses", "Smile", "Anger", "Sadness", "Disgust", "Fear", "Surprise", "Neutral", "Laugh", "Happy", "Confused", "Scream".
++Eye	int	O	Eye status, 0: not detected, 1: close eye, 2: open eye.
++Mouth	int	O	Mouth status, 0: not detected, 1: close mouth, 2: open mouth.

++Mask	int	O	Mask status, 0: not detected, 1: not wearing mask, 2: wearing mask.
++Beard	int	O	Beard status, 0: not detected, 1: no beard, 2: has beard.
++Glass	Int	O	Glasses status, 0: unknown, 1: not wearing, 2: normal Glasses, 3: sun glasses, 4: black frame glasses.

[Example]

Event	--<boundary>
	Content-Type: text/plain Content-Length: <length> Events[0].EventBaseInfo .Code=HumanTrait Events[0].EventBaseInfo .Action=Pulse Events[0].EventBaseInfo .Index=0 Events[0].HumanAttributes.BoundingBox[0]=1341 Events[0].HumanAttributes.BoundingBox[1]=2451 Events[0].HumanAttributes.BoundingBox[2]=4513 Events[0].HumanAttributes.BoundingBox[3]=4135 Events[0].HumanAttributes.Sex=Man Events[0].HumanAttributes.Age=30 Events[0].HumanAttributes.CoatColor=White Events[0].HumanAttributes.CoatType=1 Events[0].HumanAttributes.TrousersColor=Black Events[0].HumanAttributes.TrousersType=1 Events[0].HumanAttributes.HasHat=1 Events[0].HumanAttributes.HasBag=2 Events[0].FaceAttributes.BoundingBox[0]=1341 Events[0].FaceAttributes.BoundingBox[1]=2451 Events[0].FaceAttributes.BoundingBox[2]=4513 Events[0].FaceAttributes.BoundingBox[3]=4135 Events[0].FaceAttributes.Sex=Man Events[0].FaceAttributes.Age=30 Events[0].FaceAttributes.Feature[0]=Smile Events[0].FaceAttributes.Eye=2 Events[0].FaceAttributes.Mouth=1 Events[0].FaceAttributes.Glass=1 --<boundary> Content-Type: image/jpeg Content-Length: <image size> <Jpeg image data> --<boundary>

9.1.6 [Event] CrossLineDetection

Table 9-6

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event
-------	--

Description	When detect some object cross the line, send this event.		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
Object	object	R	The object that cross the line.
+BoundingBox	Array<int>	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
Objects	array<object>	O	If detect several object, store in this array.
+BoundingBox	Array<int>	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
DetectLine	Array<Array<int>>	R	The detection line, the first array is point list, max item is 20, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.
Direction	string	O	The crossline direction, can be : “LeftToRight”, “RightToLeft”, “Any”.
[Example]			
Event	<pre>--<boundary> Content-Type: text/plain Content-Length: <length> Code=CrossLineDetection;action=pulse;index=0;data={ "Object": { "BoundingBox": [2992,1136,4960,5192] }, "DetectLine": [[1292,3469], [6535,3373], ...], "Direction" : "LeftToRight" } --<boundary></pre>		

9.1.7 [Event] CrossRegionDetection

Table 9-7

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When detect some object cross the region, send this event		
[Event Params] (JSON format)			
Name	Type	R/ O	Param Description
Object	object	R	The object that cross the region.
+BoundingBox	Array<int>	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left-top point, x's value of right—bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
Objects	array<object>	O	If detect several object, store in this array.

+BoundingBox	Array<int>	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
DetectRegion	Array<Array<int>>	R	The detection region, the first array is point list, max item is 20, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.
Action	string	R	The cross action, can be : "Appear", "Disappear", "Cross", "Inside"
Direction	string	O	The cross direction, valid on if the "Action" is "Cross", can be : "Enter", "Leave", "Both".

[Example]

Event	--<boundary> Content-Type: text/plain Content-Length: <length> Code=CrossRegionDetection;action=pulse;index=0;data={ "Object": { "BoundingBox": [2992,1136,4960,5192] }, "DetectRegion": [[1292,3469], [6535,3373], ...], "Action" : "Cross", "Direction" : "Enter" } --<boundary>
-------	---

9.1.8 [Event] Queue Stay Detection

Table 9-8

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When detect the queue stay time too long, send this event		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
Object	object	O	The object that was stay.
+BoundingBox	Array<int>	O	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.
Objects	array<object>	O	If detect several object, store in this array.
+BoundingBox	Array<int>	O	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.
DetectRegion	Array<Array<int>>	O	The detection region, the first array is point list, max item is 20, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.
AreaID	int	O	The area id, begin from 1, if omit, means single area.

PresetID	int	O	The preset id, Valid ID starts from 1; 0 means meaningless, not involved. if omit, means preset is unknown.
[Example]			
Event	<pre>--<boundary> Content-Type: text/plain Content-Length: <length> Code=QueueStayDetection;action=start;index=0;data={ "Object": { "BoundingBox": [2992,1136,4960,5192] }, "Objects": [{ "BoundingBox": [2992,1136,4960,5192] }, { "BoundingBox": [4392,4136,6960,6512] }, { ... }, ...], "DetectRegion": [[1292,3469], [6535,3373], ...], "AreaID" : 2 } --<boundary></pre>		

9.1.9 [Event] QueueNumDetection

Table 9-9

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When detect the queue people number exceed limit, send this event		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
ManList	Array<object>	R	The people info list.
+BoundingBox	Array<int>	R	The detected people bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
+Stature	int	R	The people's stature, unit is cm.
AreaID	int	O	The area id, begin from 1, if omit, means single area.
PresetID	int	O	The preset id, Valid ID starts from 1; 0 means meaningless, not involved. if omit, means preset is unknown.
[Example]			
Event	<pre>--<boundary> Content-Type: text/plain Content-Length: <length> Code=QueueNumDetection;action=pulse;index=0;data={ "ManList": [{ "BoundingBox": [2992,1136,4960,5192], "Stature": 170 }, { "BoundingBox": [4392,4136,6960,6512], "Stature": 175 }, { ... }, ...], "AreaID" : 2 } --<boundary></pre>		

9.2 Face Recognition Server

9.2.1 Create Face Group

Table 9-10

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=createGroup		
Method	GET		
Description	Create a face group.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
groupName	string	R	The face group name, max string length is 127.
groupDetail	string	O	The description detail of the face group, max string length is 255.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
groupId	string	R	The identity of the created face group, max string length is 63.
[Example]			
Request	GET http://192.168.1.250/cgi-bin/faceRecognitionServer.cgi?action=createGroup&groupName=Test1&groupDetail=ForTest1		
Response	groupId=10000		

9.2.2 Modify Face Group

Table 9-11

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=modifyGroup		
Method	GET		
Description	Modify a face group.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
groupId	string	R	The identity of the face group, max string length is 63.
groupName	string	R	The name of the the face group, max string length is 127.
groupDetail	string	O	Description detail of the face group, max string length is 255.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.250/cgi-bin/faceRecognitionServer.cgi?action=modifyGroup&groupId=10000&groupName=Test1&groupDetail=ForTest1		
Response	OK		

9.2.3 Delete Face Group

Table 9-12

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=deleteGroup		
Method	GET		
Description	Delete a face group.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
groupId	string	R	The identity of the face group, max string length is 63.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.250/cgi-bin/faceRecognitionServer.cgi?action=deleteGroup&groupId=100 00		
Response	OK		

9.2.4 Deploy Face Group

There are two ways to deploy the group. One is based on the group (putDisposition), and the another one is based on the channel (setGroup).

- Put disposition to group

Table 9-13

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=putDisposition		
Method	GET		
Description	Deploy the face group to some video channels. If the video channel has been deployed already, it will change the similary.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
groupId	string	R	The identity of the face group, max string length is 63.
list	Array<object>	R	List of disposition info.
+channel	int	R	Video channel index which starts from 1.
+similary	int	R	The threshold of the face similary, 0 — 100.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
report	array<bool>	R	Result of putting disposition for each request channel.
[Example]			
Request	GET http://192.168.1.250/cgi-bin/faceRecognitionServer.cgi?action=putDisposition&groupId=1 0000&list[0].channel=1&list[0].similary=80&list[1].channel=2&list[1].similary=70		
Response	report[0]=true report[1]=false		

- Delete some disposition from group

Table 9-14

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=deleteDisposition		
Method	GET		
Descripti	Remove the deployment of face group from some video channels.		

on			
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
groupId	string	R	The identity of the face group, max string length is 63.
channel	Array<int>	R	Video channel index which starts from 1.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
report	array<bool>	R	Result of deleting disposition for each request channel.
[Example]			
Request	GET http://192.168.1.250/cgi-bin/faceRecognitionServer.cgi?action=deleteDisposition&groupId=10000&channel[0]=1&channel[1]=2		
Response	report[0]=true report[1]=false		

- set disposition group to channel

Table 9-15

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=setGroup		
Method	GET		
Description	Deploy some face groups to one video channel. If the video channel has been deployed already, it will change the similiary. Note: This method will do an overwrite operation.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
channel	int	R	Video channel index which starts from 1.
list	Array<object>	O	List of disposition info, if not exist, remove all disposition from channel.
+groupId	int	R	The identity of the face group, max string length is 63.
+similiary	int	R	The threshold of the face similiary, 0 — 100.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.250/cgi-bin/faceRecognitionServer.cgi?action=setGroup&channel=1&list[0].groupId=10000&list[0].similiary=80&list[1].groupId=10002&list[1].similiary=75		
Response	OK		

- get disposition group from channel

Table 9-16

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=getGroup		
Method	GET		
Description	Get the Deployment about the video channel. Note: If the video channel does not deploy any group, then the response will be success with empty http body.		
[Request Params] (key=value format at URL)			

Name	Type	R/O	Param Description
channel	int	R	Video channel index which starts from 1.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
groupId	Array<int>	R	The identity of the face group, max string length is 63.
similary	Array<int>	R	The threshold of the face similary, 0 — 100.
[Example]			
Request	GET http://192.168.1.250/cgi-bin/faceRecognitionServer.cgi?action=getGroup&channel=1		
Response	groupId[0]=10001 groupId[1]=10003 groupId[2]=10006 similary[0]=80 similary[1]=75 similary[2]=85		

9.2.5 Find Face Group

Table 9-17

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=findGroup		
Method	GET		
Description	Find the face group. If the groupId is not present in the URL, it will return all the groups.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
groupId	string	O	The identity of the face group, max string length is 63.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
GroupList	Array<object>	R	The face group information list.
+groupId	string	R	The identity of the face group, max string length is 63.
+groupName	string	R	Name of the face group, max string length is 127.
+groupDetail	string	O	Description detail of the face group, max string length is 255.
+groupSize	int	R	The number of face in this face group.
+channels	Array<int>	O	Video channel index which starts from 0.
+similarity	Array<int>	O	The threshold of the face similary.
+groupType	string	O	The type of face group
+TimeSection	Array<Array<string>>	O	The time section of face group
[Example]			
Request	GET http://192.168.1.250/cgi-bin/faceRecognitionServer.cgi?action=findGroup		
Response	GroupList[0].groupId=00001 GroupList[0].groupName=Test1 GroupList[0].groupDetail=ForTest1 GroupList[0].groupSize=30 GroupList[0].channels[0]=1		

```
GroupList[0].channels[1]=2  
...  
GroupList[0].similarity[0]=80  
GroupList[0].similarity[1]=75  
...  
GroupList[0].groupType=BlackListDB  
GroupList[0].TimeSection[0][0]=1 00:00:00-23:59:59  
GroupList[0].TimeSection[0][1]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[0][2]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[0][3]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[0][4]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[0][5]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[1][0]=1 00:00:00-23:59:59  
GroupList[0].TimeSection[1][1]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[1][2]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[1][3]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[1][4]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[1][5]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[2][0]=1 00:00:00-23:59:59  
GroupList[0].TimeSection[2][1]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[2][2]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[2][3]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[2][4]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[2][5]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[3][0]=1 00:00:00-23:59:59  
GroupList[0].TimeSection[3][1]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[3][2]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[3][3]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[3][4]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[3][5]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[4][0]=1 00:00:00-23:59:59  
GroupList[0].TimeSection[4][1]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[4][2]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[4][3]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[4][4]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[4][5]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[5][0]=1 00:00:00-23:59:59  
GroupList[0].TimeSection[5][1]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[5][2]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[5][3]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[5][4]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[5][5]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[6][0]=1 00:00:00-23:59:59  
GroupList[0].TimeSection[6][1]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[6][2]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[6][3]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[6][4]=0 00:00:00-23:59:59
```

	<pre> GroupList[0].TimeSection[6][5]=0 00:00:00-23:59:59 GroupList[1].groupID=00003 GroupList[1].groupName=Test3 GroupList[1].groupDetail=ForTest3 GroupList[1].groupSize=50 GroupList[1].channels[0]=1 GroupList[1].channels[1]=2 ... GroupList[1].similarity[0]=70 GroupList[1].similarity[1]=85 ... </pre>
--	--

9.2.6 Re-Abstract Feature By Group

- Start ReAbstract

Table 9-18

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=groupReAbstract		
Method	GET		
Description	Abstract features for the groups. About the process of the re-extract, the device will use an event named " FaceFeatureAbstract " to report the process.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
groupId	Array<int>	R	The identity of the face group, max string length is 63.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
token	int	R	The identity of this operation.
[Example]			
Request	GET http://192.168.1.250/cgi-bin/faceRecognitionServer.cgi?action=groupReAbstract&groupId[0]=10000&groupId[1]=10001		
Response	token=12345		

- Stop ReAbstract

Table 9-19

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=stopGroupReAbstract		
Method	GET		
Description	Stop the abstract features process.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
token	int	R	The identity of this operation.
[Response Params] (OK)			

[Example]	
Request	GET http://192.168.1.250/cgi-bin/faceRecognitionServer.cgi?action=stopGroupReAbstract&token=12345
Response	OK

9.2.7 Add Person

Table 9-20

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=addPerson		
Method	POST		
Description	Add a person to a group.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
groupId	string	R	The identity of the face group that this person to add. Max string length is 63.
name	string	R	The person name, max string length is 63.
birthday	string	O	The person's birthday, ex: "1980-01-01".
sex	string	O	Sex, it can be "Male", "Female", "Unknown".
country	string	O	The country name, length must be 2, and value should be according to ISO3166.
province	string	O	The province name, max string length is 63.
city	string	O	The city name, max string length is 63.
certificateType	string	O	The certificate type. It can be: "IC", "Passport", "Unknown".
id	string	O	The ID of certificate type, max string length is 31.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
uid	string	R	The id for this Person, max string length is 31.
[Example]			
Request	POST http://<server>/cgi-bin/faceRecognitionServer.cgi?action=addPerson&groupId=10000&name=Rohit&birthday=1980-01-05&sex=Male&country=IND&province=XXX&city=YYY HTTP/1.1 Content-Type: image/jpeg Content-Length: <image size> <JPEG image data>		
Response	uid=0005		

9.2.8 Modify Person

Table 9-21

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=modifyPerson
Method	POST

Description	Modify a person's info. Note: If you do not want to change the image about the person, the request should not contain the image data. Note: You should provide at least one optional param to update.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
uid	string	R	The identity of the Person, max string length is 31.
groupId	string	R	The identity of the Face Group that this Person in. max string length is 63.
name	string	O	The person's name, max string length is 63.
birthday	string	O	The person's birthday, ex: "1980-01-01".
sex	string	O	Sex, it can be "Male", "Female", "Unknown".
country	string	O	The country name, length must be 2, and value should be according to ISO3166.
province	string	O	The province name, max string length is 63.
city	string	O	The city name, max string length is 63.
certificateType	string	O	The certificate type. It can be: "IC", "Passport", "Unknown".
id	string	O	The ID of certificate type, max string length is 31.
[Response Params] (OK)			
[Example]			
Request	POST http://<server>/cgi-bin/faceRecognitionServer.cgi?action=modifyPerson&uid=0005&grou pID=10000&name=Rohit&birthday=1980-01-05&sex=Male&country=IND&province= XXX&city=YYY HTTP/1.1 Content-Type: image/jpeg Content-Length: <image size> <JPEG image data>		
Response	OK		

9.2.9 Delete Person

Table 9-22

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=deletePerson		
Method	GET		
Description	Delete a person from a group.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
uid	string	R	The identity of the person, max string length is 31.
groupId	string	R	The identity of the face group that this Person in. max string length is 63.
[Response Params] (OK)			
[Example]			

Request	GET http://192.168.1.250/cgi-bin/faceRecognitionServer.cgi?action=deletePerson&uid=001&groupID=10000
Response	OK

9.2.10 Find Person

- Start to find

Table 9-23

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=startFind		
Method	GET		
Description	Start to find person in face groups. Note: the returned token will be expired after 60 seconds without any doFind call.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
condition	Object	R	Search scope condition.
+GroupID	Array<string>	R	The list of identity of the face group, max string length is 63.
person	object	O	Person condition.
+Name	string	O	Person Name, max string length is 63.
+Sex	string	O	Sex, it can be "Male", "Female", "Unknown".
+Country	string	O	Country name, length must be 2, and value should be according to ISO3166.
+Province	string	O	Province name, max string length is 63.
+City	string	O	City name, max string length is 63.
+CertificateType	string	O	Certificate Type. It can be: "IC", "Passport", "Unknown".
+ID	string	O	Person ID of CertificateType, max string length is 31.
+FeatureState	int	O	Feature State, 0:Unknown, 1:Failed, 2:OK.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
token	uint	R	Token for this search, use this token to get result and stop search.
totalCount	int	R	Result num, return -1 means still searching.
[Example]			
Request	GET http://<server>/cgi-bin/faceRecognitionServer.cgi?action=startFind&condition.GroupID[0]=10000&condition.GroupID[1]=10003&person.Sex=Male&person.Country=IND&person.FeatureState=1		
Response	token=123456789 totalCount=24		

- Get find result

Table 9-24

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=doFind
Method	GET

Description	Get one result of person's information from the search result set. Note: the returned token will be expired after 60 seconds without any doFind call.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
token	uint	R	Token for this search, use this token to get result and stop search.
index	uint	R	The index in search result, should between 0 and totalCount -1.
[Response Params] (JSON format)			
Name	Type	R/O	Param Description
person	object	R	Person condition.
+UID	string	R	The identity of the person, max string length is 31.
+GroupID	string	R	The identity of the face group that this Person in. max string length is 63.
+Name	string	R	The person name, max string length is 63.
+Sex	string	O	Sex, it can be "Male", "Female", "Unknown".
+Birthday	string	O	The person's birthday, ex: "1980-01-01".
+Country	string	O	Country name, length must be 2, and value should be according to ISO3166.
+Province	string	O	Province name, max string length is 63.
+City	string	O	City name, max string length is 63.
+CertificateType	string	O	Certificate Type, can be: "IC", "Passport", "Unknown".
+ID	string	O	Person ID of CertificateType, max string length is 31.
+FeatureState	int	O	Feature State, 0:Unknown, 1:Failed, 2:OK.
[Example]			
Request	<p>GET <a href="http://<server>/cgi-bin/faceRecognitionServer.cgi?action=doFind&token=123456789&index=0">http://<server>/cgi-bin/faceRecognitionServer.cgi?action=doFind&token=123456789&index=0</p>		
Response	<p>HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<boundary> Content-Length: <length></p> <p>--<boundary> Content-Type: text/plain Content-Length: <length></p> <p>person.UID=0005 person.GroupID=10000 person.Name=Rohit person.Birthday=1980-01-01 person.Sex=Male person.Country=IND person.Province=XXX person.City=YYY person.CertificateType=IC</p>		

	person.ID=1234567890 person.FeatureState=0 --<boundary> Content-Type: image/jpeg Content-Length: <image size> < jpeg image data ... > --<boundary>--
--	--

- Stop finding

Table 9-25

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=stopFind		
Method	GET		
Description	Stop the search session.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
token	uint	R	The token for this search, use this token to get result and stop search.
[Response Params] (OK)			
[Example]			
Request	GET http://<server>/cgi-bin/faceRecognitionServer.cgi?action=stopFind&token=123456789		
Response	OK		

9.2.11 Re-Abstract Features By Person

- Start ReAbstract

Table 9-26

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=reAbstract		
Method	GET		
Description	Abstract features for the persons. About the process of the re-extract, the device will use an event named "FaceFeatureAbstract"to report the process.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
UID	Array<int>	O	The list of identity of person, max string length is 31.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.250/cgi-bin/faceRecognitionServer.cgi?action=reAbstract&uid[0]=001&uid[1]=002		
Response	OK		

- Stop ReAbstract

Table 9-27

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=stopReAbstract
Method	GET
Description	Stop the abstract features process.
[Request Params] (None)	
[Response Params] (OK)	
[Example]	
Request	GET http://192.168.1.250/cgi-bin/faceRecognitionServer.cgi?action=stopReAbstract
Response	OK

9.2.12 [Config] Face Recognition AlarmOut Setting

- Get FaceRecognitionAlarm config

Table 9-28

URL	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=FaceRecognitionAlarm		
Method	GET		
Description	Get the Face Recognition Alarm Out Setting.		
[Request Params] (None)			
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
table	object	R	Config info table.
+FaceRecognitionAlarm	array<object>	R	Each face group has one config object in this array.
++GroupID	String	R	The face group ID, max string length is 63.
++GroupName	String	R	The face group name, max string length is 127.
++AlarmOutEnable	Bool	R	Enable AlarmOut or not.
++AlarmChannel	Array<object>	R	Each AlarmOut channel has one config object in this array.
+++AlarmRuleMask	Int	R	Alarm rule mask. • Bit 0 : recognition success • Bit 1 : recognition failed
+++AlarmOutLatCh	Int	R	Alarm out delay, unit is second, value between 1 and 300.
[Example]			
Request	GET http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=FaceRecognitionAlarm		

Response	<pre> table.FaceRecognitionAlarm[0].GroupID=0017 table.FaceRecognitionAlarm[0].GroupName=wsd table.FaceRecognitionAlarm[0].AlarmOutEnable=true table.FaceRecognitionAlarm[0].AlarmChannel[0].AlarmRuleMask=0 table.FaceRecognitionAlarm[0].AlarmChannel[0].AlarmOutLatch=5 table.FaceRecognitionAlarm[0].AlarmChannel[1].AlarmRuleMask=0 table.FaceRecognitionAlarm[0].AlarmChannel[1].AlarmOutLatch=8 ... table.FaceRecognitionAlarm[1].GroupID=0018 table.FaceRecognitionAlarm[1].GroupName=cst table.FaceRecognitionAlarm[1].AlarmOutEnable=true table.FaceRecognitionAlarm[1].AlarmChannel[0].AlarmRuleMask=0 table.FaceRecognitionAlarm[1].AlarmChannel[0].AlarmOutLatch=10 table.FaceRecognitionAlarm[1].AlarmChannel[1].AlarmRuleMask=0 table.FaceRecognitionAlarm[1].AlarmChannel[1].AlarmOutLatch=15 ... </pre>
----------	--

- Set FaceRecognitionAlarm config

URL	http://<server>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set the Face Recognition Alarm Out Setting.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
FaceRecognitionAlarm	array<object>	R	Each face group has one config object in this array.
+GroupId	String	R	The face group ID, max string length is 63.
+GroupName	String	R	The face group name, max string length is 127.
+AlarmOutEnable	Bool	R	Enable AlarmOut or not.
+AlarmChannel	Array<object>	R	Each AlarmOut channel has one config object in this array.
++AlarmRuleMask	Int	R	Alarm rule mask. ● Bit 0 : recognition success ● Bit 1 : recognition failed
++AlarmOutLatch	Int	R	Alarm out delay, seconds, value between 1 and 300.
[Response Params] (OK)			
[Example]			
Request	GET http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&FaceRecognitionAlarm[0].GroupID=0017&FaceRecognitionAlarm[0].GroupName=wsd&FaceRecognitionAlarm[0].AlarmOutEnable=true&FaceRecognitionAlarm[0].AlarmChannel[0].AlarmRuleMask=0&FaceRecognitionAlarm[0].AlarmChannel[0].AlarmOutLatch=5		
Response	OK		

9.2.13 Find Person by Picture

- Start to find

Table 9-29

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=startFindByPic		
Method	POST		
Description	Start to find person in face groups by picture. The search may last for some time, so the response may push at regular intervals until 100% Progress Note: If you want to find person in face groups by person info, see above "Find Person"API. Note: The returned token will be expired after 60 seconds without any doFind call.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
GroupID	array<string>	R	Face groups to find, max string length is 63.
Similarity	int	R	Similarity percent, 1 — 100.
MaxCandidate	int	O	Max Candidate result number.
[Response Params] (JSON format)			
Name	Type	R/O	Param Description
token	uint	R	The token of this search, use this token to get result and stop search.
progress	uint	R	Search Progress, 100 means finished.
totalCount	int	R	Result num, return -1 means still searching.
[Example]			
Request	POST http://<server>/cgi-bin/faceRecognitionServer.cgi?action=startFindByPic&GroupID[0]=00001&GroupID[1]=00003&Similarity=80&MaxCandidate=500 Content-Type: image/jpeg Content-Length: <image size> <JPEG data>		
Response	HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<boundary> Connection: closed --<boundary> Content-Type: application/json Content-Length: <length> { "token": 123456789, "progress": 20 "totalCount": -1 } --<boundary> Content-Type: application/json		

	Content-Length: <length> { "token": 123456789, "progress": 60 "totalCount": -1 } --<boundary> Content-Type: application/json Content-Length: <length> { "token": 123456789, "progress": 100 "totalCount": 350 } --<boundary>--
--	--

- Get the find result

Table 9-30

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=doFindByPic		
Method	GET		
Description	Get the find result, reply using multipart format, first part is json string to describe all candidate person, then the following parts are the person's pictures, refer by UID and GroupID in part header Content-Info. Note: the returned token will be expired after 60 seconds without any doFind call.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
token	uint	R	Token for this search, use this token to get result and stop search.
index	uint	R	The index in search result, should between 0 and totalCount -1.
count	uint	R	Number of result person to get.
[Response Params] (JSON format)			
Name	Type	R/O	Param Description
Found	int	R	Number of result person that return.
Candidates	Array<object>	R	Candidates Person.
+Person	object	R	Person Info.
++UID	string	R	System id for this Person, max string length is 31.
++GroupID	string	R	The identity of the Face Group that this Person in. max string length is 63.
++Name	string	R	Person Name, max string length is 63.
++Birthday	string	O	Birthday ex: "1980-01-01".
++Sex	string	O	Sex, it can be "Male", "Female", "Unknown".
++Country	string	O	Country name, length must be 2, value should be according to ISO3166.
++Province	string	O	Province name, max string length is 63.
++City	string	O	City name, max string length is 63.

++CertificateType	string	O	Certificate Type. It can be "IC", "Passport", or "Unknown".
++ID	string	O	Person ID of CertificateType, max string length is 31.
++FeatureState	int	O	Feature State, 0:Unknown, 1:Failed, 2:OK.
++HomeAddress	string	O	Home Address, ex: "binanRoad NO1199"
+Similarity	int	R	Similarity.
[Example]			
Request	GET http://<server>/cgi-bin/faceRecognitionServer.cgi?action=doFindByPic&token=123456789&index=0&count=10		
Response	HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<boundary> Connection: closed --<boundary> Content-Type: application/json Content-Length: <length> <pre>{ "Found": 10, "Candidates": [{ "person": { "UID": "0001", "GroupID": "001", "Name": "Rohit", "Birthday": "1980-01-05", "Sex": "Male", ... }, "Similarity": 85 }, { "person": { "UID": "0002", "GroupID": "002", "Name": "LiSi", "Birthday": "1980-01-06", "Sex": "Male", ... }, "Similarity": 80 }, { ... }, ...] }</pre>		

```

}
--<boundary>
Content-Info: UID=0001&GroupID=001
Content-Type: image/jpeg
Content-Length: <length>

< jpeg image data ... >
--<boundary>
Content-Info: UID=0002&GroupID=002
Content-Type: image/jpeg
Content-Length: <length>

< jpeg image data ... >
--<boundary>
...

```

- Stop finding

Table 9-31

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=stopFindByPic		
Method	GET		
Description	Stop the search session.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
token	uint	R	Token for this search, use this token to get result and stop search.
[Response Params] (OK)			
[Example]			
Request	GET http://<server>/cgi-bin/faceRecognitionServer.cgi?action=stopFindByPic&token=123456789		
Response	OK		

9.2.14 Find History Person by Picture

- Start to find

Table 9-32

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=startFindHistoryByPic	
Method	POST	
Description	Start to find person in capture history by picture. The search may last for some time, so the response may push at regular intervals until 100% Progress. Note: If you want to find person in capture history by person info, please refer to "mediaFileFind"API. Note: the returned token will be expired after 60 seconds without any doFind call.	

[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
Channel	int	R	Video channel index which starts from 0.
StartTime	string	R	Start time to search, ex: 2018-01-13T00:00:00Z.
EndTime	string	R	End time to search, ex: 2018-01-14T00:00:00Z.
Similarity	int	R	Similarity percent, 1 — 100.
MaxCandidate	int	O	Max Candidate result number.
[Response Params] (JSON format)			
Name	Type	R/O	Param Description
token	uint	R	Token for this search, use this token to get result and stop search.
progress	uint	R	Search Progress, 100 means finished.
totalCount	int	R	Result num, return -1 means still searching.
[Example]			
Request	POST http://<server>/cgi-bin/faceRecognitionServer.cgi?action=startFindHistoryByPic&Channe l=0&StartTime=2018-01-13T00:00:00Z&EndTime=2018-01-14T00:00:00Z&Type=All&Si milarity=80&MaxCandidate=500 Content-Type: image/jpeg Content-Length: <image size> <JPEG data>		

	HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<boundary> Connection: closed
	--<boundary> Content-Type: application/json Content-Length: <length>
	{ "token": 123456789, "progress": 20 "totalCount": -1 } --<boundary>
Response	Content-Type: application/json Content-Length: <length>
	{ "token": 123456789, "progress": 60 "totalCount": -1 } --<boundary>
	Content-Type: application/json Content-Length: <length>
	{ "token": 123456789, "progress": 100 "totalCount": 350 } --<boundary>--

- Get find result

Table 9-33

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=doFindHistoryByPic		
Method	GET		
Description	Get the find result, reply by multipart, first part is json string to describe all candidate person, then the following part is the person's picture, refer by UID in part header Content-Info. Note: the returned token will be expired after 60 seconds without any doFind call.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
token	uint	R	Token for this search, use this token to get result and stop search.
index	uint	R	The index in search result, should between 0 and totalCount -1.
count	uint	R	Number of result person to get, start from Index.
[Response Params] (JSON format)			

Name	Type	R/O	Param Description
Found	int	R	Number of result person that return.
Candidates	Array<object>	R	Candidates Person.
+Person	object	R	Person Info.
++UID	string	R	System id for this Person, max string length is 63.
++Sex	string	O	Sex. It can be "Male", "Female" or "Unknown".
++Age	int	O	Age.
++Glasses	int	O	Glasses Status, 0: all, 1: not wear, 2: wear.
+Similarity	int	R	Similarity.
+Time	string	O	Appear time of the Person, format is "2013-09-02 00:00:00".

[Example]

Request	<p>GET <a href="http://<server>/cgi-bin/faceRecognitionServer.cgi?action=doFindHistoryByPic&token=123456789&index=0&count=12">http://<server>/cgi-bin/faceRecognitionServer.cgi?action=doFindHistoryByPic&token=123456789&index=0&count=12</p>
Response	<p>HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<boundary> Connection: closed</p> <p>--<boundary> Content-Type: application/json Content-Length: <length></p> <pre>{ "Found": 12, "Candidates": [{ "person": { "UID": "0001", "Sex": "Male", "Age": 30, "Glasses": 1 }, "Similarity": 85, "Time": "2013-09-02 00:00:00" }, { "person": { "UID": "0002", "Sex": "Male", "Age": 50, "Glasses": 2 }, "Similarity": 80, "Time": "2013-09-02 00:00:00" }, { ... }, ...] }</pre>

```

}
--<boundary>
Content-Info: UID=0001
Content-Type: image/jpeg
Content-Length: <length>

< jpeg image data ... >
--<boundary>
Content-Info: UID=0002
Content-Type: image/jpeg
Content-Length: <length>

< jpeg image data ... >
--<boundary>
...

```

- Stop finding

Table 9-34

URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=stopFindHistoryByPic		
Method	GET		
Description	Stop finding.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
token	uint	R	Token for this search, use this token to get result and stop searching.
[Response Params] (OK)			
[Example]			
Request	GET http://<server>/cgi-bin/faceRecognitionServer.cgi?action=stopFindHistoryByPic&token=123456789		
Response	OK		

9.2.15 [Event] FaceDetection

Table 9-35

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When the video channel disposition with some face group, and the video channel detect a face, send this event.		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
Faces	Array<object>	R	The detected faces info.

+BoundingBox	Array<int>	R	The detected face bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.
+Sex	string	O	Sex, it can be "Man", "Woman".
+Age	int	O	Age.
+Feature	array<string>	O	Face feature, can be some of the following: "WearGlasses", "SunGlasses", "NoGlasses", "Smile", "Anger", "Sadness", "Disgust", "Fear", "Surprise", "Neutral", "Laugh", "Happy", "Confused", "Scream".
+Eye	int	O	Eye status, 0: not detected, 1: close eye, 2: open eye.
+Mouth	int	O	Mouth status, 0: not detected, 1: close mouth, 2: open mouth.
+Mask	int	O	Mask status, 0: not detected, 1: not wearing mask, 2: wearing mask.
+Beard	int	O	Beard status, 0: not detected, 1: no beard, 2: has beard.

[Example]

Event	--<boundary> Content-Type: text/plain Content-Length: </length> Code=FaceDetection;action=Start;index=0;data={ "Faces": [{ "BoundingBox": [2992,136,6960,8192], "Sex": "Man", "Age": 40, "Feature": ["WearGlasses", "Smile"], "Eye": 2, "Mouth": 1, "Mask": 1, "Beard": 2 }, {...}, ...] } --<boundary>
-------	--

9.2.16 [Event] FaceRecognition

Table 9-36

Usage	Refer to "4.4.3 Subscribe to Snapshot"for how to subscribe event		
Description	When the video channel disposition with some face group, and the video channel detect a face, after recognize in the face groups, send this event.		
[Event Params] (key=value format)			
Name	Type	R/O	Param Description
Events	array<object>	R	Event info array.
+EventBaseInfo	object	R	Base info of event.
++Code	String	R	Event Code. It should be FaceRecognition.

++Action	String	R	Event Action. It can be "Start", "Stop" or "Pulse".
++Index	int	O	The channel index relate to this event, start from 0.
+UID	String	R	The identity of the Person, max string length is 31.
+Candidates	array<object>	O	The Candidates person list.
++Person	object	O	The information of candidate person.
+++UID	string	R	The identity of the Person, max string length is 31.
+++GroupID	string	R	The identity of the Face Group that this Person in. max string length is 63.
+++Name	string	O	Person Name, max string length is 63.
+++Birthday	string	O	Birthday ex: "1980-01-01".
+++Sex	string	O	Sex, it can be "Male", "Female", "Unknown".
+++Country	string	O	Country name. The length must be 2, and value should be according to ISO3166.
+++Province	string	O	Province name, max string length is 63.
+++City	string	O	City name, max string length is 63.
+++CertificateType	string	O	Certificate Type. It can be: "IC", "Passport", "Unknown".
+++ID	string	O	Person ID of CertificateType, max string length is 31.
++Similarity	int	O	Similarity of the Candidates person and the detected person, value between 1 — 100.
+Face	object	O	The attribute information of face.
++Sex	string	O	Sex, it can be "Man", "Woman".
++Age	int	O	Age.
++Feature	array<string>	O	Face feature, can be some of the following : "WearGlasses", "SunGlasses", "NoGlasses", "Smile", "Anger", "Sadness", "Disgust", "Fear", "Surprise", "Neutral", "Laugh", "Happy", "Confused", "Scream".
++Eye	int	O	Eye status, 0: not detected, 1: close eye, 2: open eye.
++Mouth	int	O	Mouth status, 0: not detected, 1: close mouth, 2: open mouth.
++Mask	int	O	Mask status, 0: not detected, 1: not wearing mask, 2: wearing mask.
++Beard	int	O	Beard status, 0: not detected, 1: no beard, 2: has beard.

[Example]

Event	--<boundary> Content-Type: text/plain Content-Length: <length>
	Events[0].EventBaseInfo .Code=FaceRecognition
	Events[0].EventBaseInfo .Action=Pulse
	Events[0].EventBaseInfo .Index=0
	Events[0].UID=00105
	Events[0].Candidates[0].Person.UID=0012
	Events[0].Candidates[0].Person.GroupID=10000
	Events[0].Candidates[0].Person.Name=Rohit
	Events[0].Candidates[0].Person.Birthday=1980-01-02
	Events[0].Candidates[0].Person.Sex=Male

	<pre> ... Events[0].Candidates[0].Similarity=80 Events[0].Candidates[1].Person.UID=0014 Events[0].Candidates[1].Person.GroupID=10000 Events[0].Candidates[1].Person.Name=Lisi Events[0].Candidates[1].Person.Birthday=1980-01-05 Events[0].Candidates[1].Person.Sex=Male ... Events[0].Candidates[1].Similarity=75 ... Events[0].Face.Sex=Man Events[0].Face.Age=20 Events[0].Face.Feature[0]=SunGlasses Events[0].Face.Feature[1]=Smile Events[0].Face.Eye=2 Events[0].Face.Mouth=1 Events[0].Face.Mask=1 Events[0].Face.Beard=2 --<boundary> Content-Type: image/jpeg Content-Length: <image size> <Jpeg image data> --<boundary></pre>
--	--

9.2.17 [Event] FaceFeatureAbstract

Table 9-37

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When Re-Abstract Feature By Group or By Person, the abstract progress detail will send in this event.		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
Infos	Array<object>	R	Abstrace detail Info, max size is 100.
+State	String	R	Abstract state, it can be : <ul style="list-style-type: none"> • "Success" : Abstract success, • "False" : Failed to abstract; • "Process" : In Process;
+Process	int	O	The abstract progress.
+UID	string	O	The identity of the person, max string length is 31.
+GroupID	string	O	The identity of the face group, max string length is 63.
[Example]			

Event	<pre>--<boundary> Content-Type: text/plain Content-Length: <length> Code=FaceFeatureAbstract;action=Start;index=0;data={ "Infos": [{ "State": "Progress", "Progress": 30, "UID": "20005", "GroupID": "10000" }, {...}, ...] } --<boundary></pre>
-------	---

9.2.18 [Config] Face Recognition Event Handler Setting

- Get FaceRecognitionEventHandler config

Table 9-38

URL	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=FaceRecognitionEventHandler					
Method	GET					
Description	Get the Face Recognition Event Handler Setting.					
[Request Params] (None)						
[Response Params] (key=value format)						
Name	Type	R/O	Param Description			
table	object	R	Config info table.			
+FaceRecognitionEventHandler	array<object>	R	Each face group has one config object in this array.			
++GroupID	String	R	The face group ID, max string length is 63.			
++GroupName	String	R	The face group name, max string length is 127.			
++EventEnableMask	int	R	Report event mask, 0 means not to report event. <ul style="list-style-type: none">Bit 0 : recognition successBit 1 : recognition failed			
++RecordEnableMask	int	R	Record media file mask., 0 means not to record. <ul style="list-style-type: none">Bit 0 : recognition successBit 1 : recognition failed			
++RecordLatch	Int	R	Record latch time, unit is second.			
++SnapEnableMask	Int	R	Snap picture mask., 0 means not to snap picture <ul style="list-style-type: none">Bit 0 : recognition successBit 1 : recognition failed			
++MailEnableMask	Int	R	Send mail mask., 0 means not to send mail. <ul style="list-style-type: none">Bit 0 : recognition successBit 1 : recognition failed			
[Example]						

Request	<p>GET http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=FaceRecognitionEventHandler</p>																
Response	<table border="0"> <tr><td>table.FaceRecognitionEventHandler[0].GroupID=0017</td></tr> <tr><td>table.FaceRecognitionEventHandler[0].GroupName=wsd</td></tr> <tr><td>table.FaceRecognitionEventHandler[0].EventEnableMask=3</td></tr> <tr><td>table.FaceRecognitionEventHandler[0].RecordEnableMask=0</td></tr> <tr><td>table.FaceRecognitionEventHandler[0].RecordLatch=10</td></tr> <tr><td>table.FaceRecognitionEventHandler[0].SnapEnableMask=3</td></tr> <tr><td>table.FaceRecognitionEventHandler[0].MailEnableMask=0</td></tr> <tr><td>...</td></tr> <tr><td>table.FaceRecognitionEventHandler[1].GroupID=0018</td></tr> <tr><td>table.FaceRecognitionEventHandler[1].GroupName=cst</td></tr> <tr><td>table.FaceRecognitionEventHandler[1].EventEnableMask=3</td></tr> <tr><td>table.FaceRecognitionEventHandler[1].RecordEnableMask=0</td></tr> <tr><td>table.FaceRecognitionEventHandler[1].RecordLatch=10</td></tr> <tr><td>table.FaceRecognitionEventHandler[1].SnapEnableMask=3</td></tr> <tr><td>table.FaceRecognitionEventHandler[1].MailEnableMask=0</td></tr> <tr><td>...</td></tr> </table>	table.FaceRecognitionEventHandler[0].GroupID=0017	table.FaceRecognitionEventHandler[0].GroupName=wsd	table.FaceRecognitionEventHandler[0].EventEnableMask=3	table.FaceRecognitionEventHandler[0].RecordEnableMask=0	table.FaceRecognitionEventHandler[0].RecordLatch=10	table.FaceRecognitionEventHandler[0].SnapEnableMask=3	table.FaceRecognitionEventHandler[0].MailEnableMask=0	...	table.FaceRecognitionEventHandler[1].GroupID=0018	table.FaceRecognitionEventHandler[1].GroupName=cst	table.FaceRecognitionEventHandler[1].EventEnableMask=3	table.FaceRecognitionEventHandler[1].RecordEnableMask=0	table.FaceRecognitionEventHandler[1].RecordLatch=10	table.FaceRecognitionEventHandler[1].SnapEnableMask=3	table.FaceRecognitionEventHandler[1].MailEnableMask=0	...
table.FaceRecognitionEventHandler[0].GroupID=0017																	
table.FaceRecognitionEventHandler[0].GroupName=wsd																	
table.FaceRecognitionEventHandler[0].EventEnableMask=3																	
table.FaceRecognitionEventHandler[0].RecordEnableMask=0																	
table.FaceRecognitionEventHandler[0].RecordLatch=10																	
table.FaceRecognitionEventHandler[0].SnapEnableMask=3																	
table.FaceRecognitionEventHandler[0].MailEnableMask=0																	
...																	
table.FaceRecognitionEventHandler[1].GroupID=0018																	
table.FaceRecognitionEventHandler[1].GroupName=cst																	
table.FaceRecognitionEventHandler[1].EventEnableMask=3																	
table.FaceRecognitionEventHandler[1].RecordEnableMask=0																	
table.FaceRecognitionEventHandler[1].RecordLatch=10																	
table.FaceRecognitionEventHandler[1].SnapEnableMask=3																	
table.FaceRecognitionEventHandler[1].MailEnableMask=0																	
...																	

- Set FaceRecognitionEventHandler config

URL	http://<server>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set the Face Recognition Event Handler Setting.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
FaceRecognitionAlarm	array<object>	R	Each face group has one config object in this array.
+GroupId	String	R	The face group ID, max string length is 63.
+GroupName	String	R	The face group name, max string length is 127.
+EventEnableMask	int	R	Report event mask, 0 means not to report event. <ul style="list-style-type: none">Bit 0 : recognition successBit 1 : recognition failed
+RecordEnableMask	int	R	Record media file mask., 0 means not to record. <ul style="list-style-type: none">Bit 0 : recognition successBit 1 : recognition failed
+RecordLatch	Int	R	Record latch time, unit is second.
+SnapEnableMask	Int	R	Snap picture mask., 0 means not to snap picture <ul style="list-style-type: none">Bit 0 : recognition successBit 1 : recognition failed
+MailEnableMask	Int	R	Send mail mask., 0 means not to send mail. <ul style="list-style-type: none">Bit 0 : recognition successBit 1 : recognition failed
[Response Params] (OK)			
[Example]			

Request	GET http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&FaceRecognitionEventHandler[0].GroupID=0017&FaceRecognitionEventHandler[0].GroupName=wsd&FaceRecognitionEventHandler[0].EventEnableMask=3&FaceRecognitionEventHandler[0].RecordEnableMask=0&FaceRecognitionEventHandler[0].RecordLatch=10&FaceRecognitionEventHandler[0].SnapEnableMask=3&FaceRecognitionEventHandler[0].MailEnableMask=0
Response	OK

9.2.19 [Config] Face-ID Recognition Threshold

Template	http://<server>/cgi-bin/configManager.cgi?action=setConfig&CitizenPictureCompareRule.Threshold=60			
Method	GET			
Parameter Format	key=value format in URL			
Parameter	Type	Required	Description	Example
Threshold	uint8	Yes	Face-ID comparison threshold [1, 100]	60
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&CitizenPictureCompareRule.Threshold=60			

Parameter Format	OK at body			
Parameter	Type	Required	Description	Example
Example	OK			
OK				

9.2.20 Export Face Database

The exported data is binary.

Request URL	http://<server>/cgi-bin/api/FaceLiblmExport/export			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
groupId	char[64]	R	Person group ID	"10"
Password	char[64]	O	Unzip password	"abcd"
Request Example				
{ "groupID" : "10", "password": "abcd" }				

Response Params (binary in body)

Name	Type	R/O	Description	Example
Response Example				
HTTP/1.1 200 OK				
NTP Server Device/1.0				
Content-Type: application/octet-stream				
Content-Length: <length>				
< binary data>				

9.2.21 Importing Face Database

The imported data is binary.

Request URL	http://<server>/cgi-bin/api/FaceLibImExport/import			
Method	POST			
Request Params (multipart in body)				
Name	Type	R/O	Description	Example
password	char[64]	O	Unzip password	"abcd"
Request Example				
POST http://<server>/cgi-bin/api/FaceLibImExport/import	HTTP/1.1			
User-Agent: client/1.0				
Content-Type: multipart/x-mixed-replace; boundary=<boundary>				
--<boundary>				
Content-Type: application/json				
Content-Length: 20				
{				
"password": "abcd"				
}				
--<boundary>				
Content-Type: application/octet-stream				
Content-Length: 800				
< binary data>				
--<boundary>				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				
{}				

9.3 People Counting

9.3.1 Get Summary

Get summary information of people statistics in video.

Request URL	http://<server>/cgi-bin/videoStatServer.cgi?action=getSummary		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	Video channel index which starts from 1, default is 1.
Request Example			
http://192.168.1.250/cgi-bin/videoStatServer.cgi?action=getSummary&channel=1			

Response Params (key=value format in body)			
Name	Type	R/O	Description
summary	object	R	Summary information of people statistics in video .
+Channel	int	R	Video channel index which starts from 0.
+RuleName	string	R	Rule type, it can be: ● "ManNumDetection": count the people num in region, and the detail stat is in "InsideSubtotal" param. ● "NumberStat": count the people entering and leaving the region, and the detail stat is in "EnteredSubtotal" and "ExitedSubtotal"param.
+EnteredSubtotal	object	O	People enter stat.
++Total	int	R	Total enter num.
++Today	int	R	Today enter num.
++Hour	int	R	This hour enter num.
++TotalInTimeSection	int	O	Today enter num after call clearSectionStat.
+ExitedSubtotal	object	O	People leave stat.
++Total	int	R	Total leave num.
++Today	int	R	Today leave num.
++Hour	int	R	This hour leave num.
++TotalInTimeSection	int	O	Today leave num after call clearSectionStat.
+PassedSubtotal	object	O	People pass stat.
++Total	int	R	Total pass num.
++Today	int	R	Today pass num.
++Hour	int	R	This hour pass num.
++TotalInTimeSection	int	O	Today pass num after call clearSectionStat

+EnteredDupSubtotal	object	O	People duplicate enter stat.	
++Total	int	R	Total duplicate enter num.	14
++Today	int	R	Today duplicate enter num.	0
++Hour	int	R	This hour duplicate enter num.	0
++TotalInTimeSection	int	O	Today duplicate enter num after call clearSectionStat	0
+ExitedDupSubtotal	object	O	People duplicate leave stat.	
++Total	int	R	Total duplicate leave num.	32
++Today	int	R	Today duplicate leave num.	0
++Hour	int	R	This hour duplicate leave num.	0
++TotalInTimeSection	int	O	Today duplicate leave num after call clearSectionStat	0
+InsideSubtotal	object	O	People inside region stat.	
++Total	int	R	Today inside num.	65
++ManStayStat	array<object>	O	The entering and leaving stat of people that leaved.	
+++EnterTime	string	O	People enter time.	2012-01-04 00:00:00
+++ExitTime	string	O	People leave time.	2012-01-04 00:00:45

Response Example

```

summary.Channel=0
summary.RuleName=NumberStat
summary.EnteredSubtotal.Today=0
summary.EnteredSubtotal.Total=14
summary.EnteredSubtotal.TotalInTimeSection=0
summary.ExitedSubtotal.Today=0
summary.ExitedSubtotal.Total=32
summary.ExitedSubtotal.TotalInTimeSection=0
summary.PassedSubtotal.Hour=0
summary.PassedSubtotal.Today=0
summary.PassedSubtotal.Total=142
summary.PassedSubtotal.TotalInTimeSection=0
summary.EnteredDupSubtotal.Hour=0
summary.EnteredDupSubtotal.Today=0
summary.EnteredDupSubtotal.Total=0
summary.EnteredDupSubtotal.TotalInTimeSection=0
summary.EnteredSubtotal.TotalInTimeSection=17
summary.ExitedDupSubtotal.Hour=0
summary.ExitedDupSubtotal.Today=0
summary.ExitedDupSubtotal.Total=0
summary.ExitedDupSubtotal.TotalInTimeSection=0
summary.InsideSubtotal.Total=65
summary.InsideSubtotal.ManStayStat[0].EnterTime=2012-01-04 00:00:00

```

```

summary.InsideSubtotal.ManStayStat[0].ExitTime=2012-01-04 00:00:45
summary.InsideSubtotal.ManStayStat[1].EnterTime=2012-01-04 00:00:00
summary.InsideSubtotal.ManStayStat[1].ExitTime=2012-01-04 00:00:45

```

9.3.2 Query the Count of People

- Start to find

Start to find video stat info, in response, there is a token for further info finding process, and there is a totalCount shows how many data count(s).

Request URL	http://<server>/cgi-bin/videoStatServer.cgi?action=startFind		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	Video channel index which starts from 1, default is 1.
condition	object	R	Find condition.
+StartTime	string	R	Find time range start
+EndTime	string	R	Find time range end
+Granularity	string	R	The information granularity returned by the query requirements. The range is { Hour, Day, Week, Month, Season, Year } (Note: Most devices only support Hour, Day, Week.)
+RuleType	string	O	Rule type, it can be: NumberStat , ManNumDetection . If omit, default is NumberStat
+MinStayTime	int	O	Valid when ruleType is ManNumDetection , report people stay over this minimal time.
+PlanID	int	O	The plan id, only valid for dome camera.
+PtzPresetId	int	O	The ptz preset index which starts from 1, only valid for dome camera. Note: new device should use PtzPresetId instead of PlanID.
+AreaID	Array<int>	O	The area id which starts from 1, max array size is 20.
+OtherRule	string	O	The other search rule, can be : AverageStayTime
Request Example			
http://192.168.1.250/cgi-bin/videoStatServer.cgi?action=startFind&channel=1&condition.StartTime=2011-01-01%2012:00:00&condition.EndTime=2011-01-10%2012:00:00&condition.Granularity=Hour&condition.RuleType=NumberStat&condition.MinStayTime=20&condition.AreaID[0]=2&condition.AreaID[1]=3			

Response Params (key=value format in body)			
Name	Type	R/O	Description
token	int	R	Token for this search, use this token to get result and stop search.

totalCount	int	R	Number of find result.	56
Response Example				
token=12345				
totalCount=56				

- Get the find result

Get the find result of Video Stat info with channel, token, begin Number and count.

Request URL	http://<server>/cgi-bin/videoStatServer.cgi?action=doFind			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	Video channel index which starts from 1, default is 1. NOTE: must be the same as startFind.	1
token	int	R	Token for this search, use this token to get result and stop searching.	12345
beginNumber	int	R	The start count. It must be between 0 and totalCount -1.	0
count	int	R	The count of info for this query.	20
Request Example				
http://192.168.1.250/cgi-bin/videoStatServer.cgi?action=doFind&channel=1&token=12345&beginNumber=0&count=20				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
found	int	R	Number of result that return.	20
info	array<object>	R	Result information of video Stat.	
+Channel	int	R	Video channel index which starts from 0.	0
+PlanID	int	O	The plan id, only valid for dome camera.	2
+PtzPresetId	int	O	The ptz preset index which starts from 1, only valid for dome camera.	1
+AreaID	int	O	The area id which starts from 1.	3
+AverageStayTime	int	O	The average stay time, only valid when startFind with OtherRule param's value is "AverageStayTime".	20
+RuleName	string	R	Rule name, it can be: • " ManNumDetection• "NumberStat<td>NumberStat</td>	NumberStat
+StartTime	string	O	Find time range start	2012-01-04 00:00:00
+EndTime	string	O	Find time range end	2012-01-04

				00:02:00
+EnteredSubtotal	int	O	Total enter num.	14
+ExitedSubtotal	int	O	Total leave num.	5
+InsideSubtotal	int	O	Total inside num.	65
+PassedSubtotal	int	O	Total pass num.	14
+EnteredDupSubtotal	int	O	Total duplicate entered num.	14
+ExitedDupSubtotal	int	O	Total duplicate leave num.	5

Response Example

```

found=20
info[0].Channel=0
info[0].AreaID=2
info[0].RuleName=NumberStat
info[0].StartTime=2012-03-14 00:00:00
info[0].EndTime=2012-04-14 00:00:00
info[0].EnteredSubtotal=14
info[0].ExitedSubtotal=5
info[0].InsideSubtotal=65
info[0].PassedSubtotal=0
info[0].EnteredDupSubtotal=0
info[0].ExitedDupSubtotal=0
info[1].Channel=0
info[1].AreaID=3
info[1].RuleName=NumberStat
info[1].StartTime=2012-03-14 00:00:00
info[1].EndTime=2012-04-14 00:00:00
info[1].EnteredSubtotal=14
info[1].ExitedSubtotal=5
info[1].InsideSubtotal=65
info[1].PassedSubtotal=0
info[1].EnteredDupSubtotal=0
info[1].ExitedDupSubtotal=0

```

...

- Stop the searching session

Stop query video stat by channel and token.

Request URL	http://<server>/cgi-bin/videoStatServer.cgi?action=stopFind		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	Video channel index which starts from 1, default is 1. NOTE: must be the same as startFind.
token	int	R	Token for this search, use this token to get result and stop searching.

Request Example

```
GET http://192.168.1.250/cgi-bin/videoStatServer.cgi?action=stopFind&channel=1&token=12345
```

Response Params (OK in body)**Response Example**

OK

9.3.3 Clear the People Count Information

Clear the people count information.

Request URL	http://<server>/cgi-bin/videoStatServer.cgi?action=clearSectionStat		
--------------------	---	--	--

Method	GET		
---------------	-----	--	--

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
channel	int	O	Video channel index which starts from 1, default is 1.	1

Request Example

```
GET http://192.168.1.250/cgi-bin/videoStatServer.cgi?action=clearSectionStat&channel=1
```

Response Params (OK in body)**Response Example**

OK

9.3.4 Subscribe the People Count Information

Subscribe the people count information.

Request URL	http://<server>/cgi-bin/videoStatServer.cgi?action=attach		
--------------------	---	--	--

Method	GET		
---------------	-----	--	--

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
channel	int	O	Video channel index which starts from 1, default is 1.	1
heartbeat	int	O	Send heartbeat interval, range is [1, 60],unit is second. If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client, the heartbeat message is an string "Heartbeat". If this parameter is not present, its default value is 60.	5

Request Example

```
http://192.168.1.250/cgi-bin/videoStatServer.cgi?action=attach&channel=1&heartbeat=5
```

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
summary	object	R	Summary information of video Stat.	

+Channel	int	R	Video channel index which starts from 0.	0
+RuleName	string	R	Rule type, it can be: <ul style="list-style-type: none"> • "ManNumDetection": Count people num in region, detail stat is in "InsideSubtotal" param. • "NumberStat 	NumberStat
+EnteredSubtotal	object	O	People enter stat.	
++Total	int	R	Total enter num.	14
++Today	int	R	Today enter num.	2
++Hour	int	R	This hour enter num.	1
++TotalInTimeSection	int	O	Today enter num after call clearSectionStat.	0
+ExitedSubtotal	object	O	People leave stat.	
++Total	int	R	Total leave num.	32
++Today	int	R	Today leave num.	3
++Hour	int	R	This hour leave num.	2
++TotalInTimeSection	int	O	Today leave num after call clearSectionStat.	0
+PassedSubtotal	object	O	People pass stat.	
++Total	int	R	Total pass num.	32
++Today	int	R	Today pass num.	0
++Hour	int	R	This hour pass num.	0
++TotalInTimeSection	int	O	Today pass num after call clearSectionStat	0
+EnteredDupSubtotal	object	O	People duplicate enter stat.	
++Total	int	R	Total duplicate enter num.	14
++Today	int	R	Today duplicate enter num.	0
++Hour	int	R	This hour duplicate enter num.	0
++TotalInTimeSection	int	O	Today duplicate enter num after call clearSectionStat	0
+ExitedDupSubtotal	object	O	People duplicate leave stat.	
++Total	int	R	Total duplicate leave num.	32
++Today	int	R	Today duplicate leave num.	0
++Hour	int	R	This hour duplicate leave num.	0
++TotalInTimeSection	int	O	Today duplicate leave num after call clearSectionStat	0
+InsideSubtotal	object	O	People inside region stat.	
++Total	int	R	Today inside num.	65
++ManStayStat	array<object>	O	The entering and leaving stat of people that leaved.	

+++EnterTime	string	O	People enter time	2012-01-04 00:00:00
+++ExitTime	string	O	People leave time	2012-01-04 00:00:45

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: closed

--<boundary>

Content-Type: text/plain

Content-Length: </length>

summary.Channel=0

summary.RuleName=NumberStat

summary.EnteredSubtotal.Today=2

summary.EnteredSubtotal.Total=14

summary.EnteredSubtotal.TotalInTimeSection=1

summary.ExitedSubtotal.Today=3

summary.ExitedSubtotal.Total=32

summary.ExitedSubtotal.TotalInTimeSection=2

summary.PassedSubtotal.Hour=0

summary.PassedSubtotal.Today=0

summary.PassedSubtotal.Total=142

summary.PassedSubtotal.TotalInTimeSection=0

summary.EnteredDupSubtotal.Hour=0

summary.EnteredDupSubtotal.Today=0

summary.EnteredDupSubtotal.Total=0

summary.EnteredDupSubtotal.TotalInTimeSection=0

summary.EnteredSubtotal.TotalInTimeSection=17

summary.ExitedDupSubtotal.Hour=0

summary.ExitedDupSubtotal.Today=0

summary.ExitedDupSubtotal.Total=0

summary.ExitedDupSubtotal.TotalInTimeSection=0

summary.InsideSubtotal.Total=65

summary.InsideSubtotal.ManStayStat[0].EnterTime=2012-01-04 00:00:00

summary.InsideSubtotal.ManStayStat[0].ExitTime=2012-01-04 00:00:45

summary.InsideSubtotal.ManStayStat[1].EnterTime=2012-01-04 00:00:00

summary.InsideSubtotal.ManStayStat[1].ExitTime=2012-01-04 00:00:45

--<boundary>

Content-Type: text/plain

Content-Length: 11

Heartbeat

--<boundary>

Content-Type: text/plain

Content-Length: <length>

```
summary.Channel=0
summary.RuleName=NumberStat
summary.EnteredSubtotal.Today=2
summary.EnteredSubtotal.Total=14
...
...
```

9.3.5 Clear statistics in time section

Table 9-39

URL	http://<server>/cgi-bin/videoStatServer.cgi?action=clearSectionStat[&AreaID=<AreaID>][&PtzPresetId=<PtzPresetId>]		
Method	GET		
Description	Clear statistics in time section		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
channel	int	O	Video channel index which starts from 1, default is 1.
AreaID	int	O	The area index which starts from 1, valid for multiple area device, if omit means clear all area statistics.
PtzPresetId	int	O	The ptz preset index which starts from 1.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.250/cgi-bin/videoStatServer.cgi?action=clearSectionStat&AreaID=2		
Response	OK		

9.3.6 [Config] Video Widget Number Status

People Counting Overlay OSD Configuration Parameters

Config Data Params				
Name	Type	R/O	Description	Example
VideoWidgetNumberStat	object[]	R	Array, one element for each channel, and the array index is the channel number, starting from 0.	
+EncodeBlend	bool	R	Whether to overlay it to the main stream video encoding	true
+ShowEnterNum	bool	R	Whether to display the number of people entering	true
+ShowExitNum	bool	R	Whether to display the number of people leaving	true
+TextAlign	int	R	Text alignment mode: 0: Left alignment 2: Right alignment	0

Please refer to "4.2.1 Obtaining and Setting Configuration" to obtain and set the configuration. The specific examples are as follows:

Get Config Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=VideoWidgetNumberStat
```

Get Config Response Example

```
table.VideoWidgetNumberStat[0].EncodeBlend=true
table.VideoWidgetNumberStat[0].ShowEnterNum=true
table.VideoWidgetNumberStat[0].ShowExitNum=true
table.VideoWidgetNumberStat[0].TextAlign=0
...
```

Set Config Request Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoWidgetNumberStat[0].EncodeBl
end=true&VideoWidgetNumberStat[0].ShowEnterNum=true&VideoWidgetNumberStat[0].ShowExitNum
=true&VideoWidgetNumberStat[0].TextAlign=0
```

Set Config Response Example

OK

9.3.7 [Event] NumberStat

Table 9-40

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When people number triggers the rule, send this event.		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
Number	int	R	Total number in detect region.
EnteredNumber	int	R	Total number that enter the detect region.
ExitedNumber	int	R	Total number that leave the detect region.
PassedNumber	int	O	Total pass number in detect region.
EnteredDupNum ber	int	O	Total duplicate number that enter the detect region.
ExitedDupNum ber	int	O	Total duplicate number that leave the detect region.
Type	string	R	Number overrun type, it can be: "EnterOver", "ExitOver", "InsideOver", "PassOver".
AreaID	int	O	The area id, begin from 1, if omit, means single area.
PresetID	int	O	The preset id, if omit, means preset is unknown.
[Example]			
Event	<pre>--<boundary> Content-Type: text/plain Content-Length: <length> Code=NumberStat;action=Start;index=0;data={ "Number": 120, "EnteredNumber": 180, "ExitedNumber": 60, "PassedNumber": 60,</pre>		

```

        "EnteredDupNumber": 180,
        "ExitedDupNumber": 60,
        "Type": "EnterOver",
        "Area": 2
    }
--<boundary>

```

9.3.8 [Event] ManNumDetection

Table 9-41

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When people number triggers the rule, send this event.		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
ManList	Array<object>	R	The people info list.
+BoundingBox	Array<int>	R	The detected people bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
+Stature	int	R	The people's stature, unit is cm.
AreaID	int	O	The area id, begin from 1, if omit, means single area.
PresetID	int	O	The preset id, if omit, means preset is unknown.
[Example]			
Event	--<boundary> Content-Type: text/plain Content-Length: <length> Code=ManNumDetection;action=pulse;index=0;data={ "ManList": [{ "BoundingBox": [2992,1136,4960,5192], "Stature": 170 }, { "BoundingBox": [4392,4136,6960,6512], "Stature": 175 }, {...}, ...], "AreaID" : 2 } --<boundary>		

9.3.9 [Event] CrowdDetection

Table 9-42

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When crowd density overrun, send this event.		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description

CrowdList	Array<object>	O	The global crowd density overrun list.
+Center	Array<int>	R	The center point, must be two int, means x and y value, coordinate remap to 0 — 8192.
+Radius	int	R	The radius length.
RegionList	Array<object>	O	The people num overrun region list.
+RegionID	int	R	The region index.
+PeopleNum	int	R	The people count in region.

[Example]

	--<boundary> Content-Type: text/plain Content-Length: <length>
Event	Code=CrowdDetection;action=start;index=0;data={ "CrowdList": [{ "Center" : [5734,2377], "Radius" : 10}, ...{}], "RegionList" : [{ "RegionID" : 0, "PeopleNum" : 100 },...{}] } --<boundary>

9.4 Heat Map

9.4.1 Get Heat Map Information

Get heat map statistics by time

Request URL	http://<server>/cgi-bin/heatMap.cgi?action=getPicByTime			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	Video channel number, starting from 1,	1
StartTime	char[32]	R	Heat Map Start time, 24 h: yyyy-MM-dd HH:mm:ss.	"2015-08-20 00:00:00"
EndTime	char[32]	R	Heat Map End time, 24 h: yyyy-MM-dd HH:mm:ss.	"2015-08-21 23:59:59"
PtzPresetId	int	O	Preset, starting from 1	1
Request Example				
http://192.168.1.250/cgi-bin/heatMap.cgi?action=getPicByTime&channel=1&StartTime=2015-08-20%20 00:00:00&EndTime=2015-08-21%2023:59:59&PtzPresetId=1				

Response Params (binary data in body)

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: application/octet-stream

Content-Length: <length>

<heatmap data>

Appendix: Format of the heat map

0	1	2	3	4	...	15	16	17	18	...
Width	Height		Reserved			Data: every byte symbolize a pixel				

9.4.2 Get People Heat Map Information

Table 9-43

URL	http://<server>/cgi-bin/videoStatServer.cgi?action=getHeatMap							
Method	GET							
Description	Get People Heat Map Information							
[Request Params] (key=value format in URL)								
Name	Type	R/O	Param Description					
channel	int	O	Video channel index which starts from 1, default is 1.					
PlanID	int	O	The plan id, only valid for dome camera.					
StartTime	string	R	The start time, ex: "2010-05-12 20:00:00"					
EndTime	string	R	The end time, ex: "2010-05-12 22:00:00"					
HeatMapType	string	R	The heat map type, it can be: "AverageStayTime" : average stay time heat map "HumanStatistics" : human statistics heat map "HumanTrack" : human track heat map					
[Response Params] (binary data)								
[Example]								
Request	GET http://192.168.1.250/cgi-bin/videoStatServer.cgi?action=getHeatMap&channel=1&PlanID=2&StartTime=2010-05-12%2020:00:00&EndTime=2010-05-12%2022:00:00&HeatMapType=AverageStayTime							
Response	HTTP/1.1 200 OK Server: Device/1.0 Content-Type: application/octet-stream Content-Length: <length> <heat map binary data>							

Appendix A: The heat map binary data format when HeatMapType is AverageStayTime.

octet	0	1	2	3	4~7	8~11	12~15
value	version	channel	present	reserved	total data length	total lines	lines in this packet
octet	16~31						
value	reserved						
octet	32~35			36~39		40~43	44~44+4* (N-1)
value	line number			data num in this line		data 1	data N

octet	44+4* (N-1) +1~ 44+4* (N-1) +4			
value	next line number				data num in next line	data 1	data N

Note: If “lines in this packet” is 0, then all data has been sent, and connection will be closed.

Appendix B: The heat map binary data format when HeatMapType is HumanStatistics.

octet	0	1	2	3	4~7	8~11	12~15
value	vers ion	cha nnel	pres et	reserv ed	total data length	total lines	lines in this packet
octet	16~31						
value	reserved						
octet	32~35			36~39		40~43	44~44+4* (N-1)
value	line number			data num in this line		data 1	data N
octet	44+4* (N-1) +1~ 44+4* (N-1) +4		
value	next line number			data num in next line		data 1	data N

Note: If “lines in this packet” is 0, then all data has been sent, and connection will be closed.

Appendix C: The heat map binary data format when HeatMapType is HumanTrack.

octet	0	1	2	3	4~7	8~11	12~15
value	versi on	chann el	pres et	reser ved	total data length	total tracks	tracks in this packet
octet	16~31						
value	reserved						
octet	32~35			36~39		40~43	44~44+4* (N-1)
value	Object ID			track point num in this object		track point 1	track point N
octet	44+4* (N-1) +1~ 44+4* (N-1) +4		
value	next object ID			track point num in next object		track point 1	track point N

- Note:** If “tracks in this packet” is 0, then all data has been sent, and connection will be closed.

9.4.3 Subscribe People Realtime Trace Information

Table 9-44

URL	http://<server>/cgi-bin/videoStatServer.cgi?action=attachRealTraceProc
Method	GET
Description	Subscribe the people realtime trace information.

[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
channel	int	O	Video channel index which starts from 1, default is 1.
heartbeat	int	O	Send heartbeat interval, range is [1, 60], unit is second. If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client, the heartbeat message are "Heartbeat". If this parameter is not present, its default value is 60.
[Response Params] (multipart binary data)			
[Example]			
Request	GET http://192.168.1.250/cgi-bin/videoStatServer.cgi?action=attachRealTraceProc&channel=1&heartbeat=5		
Response	<p>HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<boundary> Connection: closed</p> <p>--<boundary> Content-Type: application/octet-stream Content-Length: <length></p> <p><realtime trace binary data> --<boundary> Content-Type: text/plain Content-Length: 11</p> <p>Heartbeat -<boundary> Content-Type: application/octet-stream Content-Length: <length></p> <p><realtime trace binary data> --<boundary> ...</p>		

- **Appendix A:** The realtime trace binary data format.

octet	0	1	2	3	4~7	8~11	12~15
value	version	chan nel	pres et	reser ved	total data length	reserved	object num in this packet
octet	16~31						
value	reserved						
octet	32~35		36~37	38~39	40~43	44~47	
value	object ID 1		object coord x,	object coord y	current time in utc	reserved	

octet	48~51		52~53	54~55	56~57	58~61
value	object ID 2		object coord x	object coord y	current time in utc	reserved

9.4.4 Get People History Trace Information

Table 9-45

URL	http://<server>/cgi-bin/videoStatServer.cgi?action=getHistoryTrace							
Method	GET							
Description	Get people history trace information.							
[Request Params] (key=value format in URL)								
Name	Type	R/O	Param Description					
channel	int	O	Video channel index which starts from 1, default is 1.					
PlanID	int	O	The plan id, only valid for dome camera.					
StartTime	string	R	The start time, ex: "2010-05-12 20:00:00"					
EndTime	string	R	The end time, ex: "2010-05-12 22:00:00"					
[Response Params] (multipart binary data)								
[Example]								
Request	GET http://192.168.1.250/cgi-bin/videoStatServer.cgi?action=getHistoryTrace&channel=1&PlanID=2&StartTime=2010-05-12%2020:00:00&EndTime=2010-05-12%2022:00:00							
Response	<p>HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<boundary> Connection: closed</p> <p>--<boundary> Content-Type: application/octet-stream Content-Length: <length></p> <p><history trace binary data> --<boundary> Content-Type: application/octet-stream Content-Length: <length></p> <p><history trace binary data> --<boundary> ...</p>							

- Appendix A: The history trace binary data format.

octet	0	1	2	3	4~7	8~11	12~15
value	version	channel	present	reserved	total data length	total object num	object num in this packet
octet	16~31						
value	reserved						

octet	32~35	36~37	38~39	40~43	44~47
value	object ID 1	object coord x	object coord y	current time in utc	reserved
octet	48~51	52~53	54~55	56~57	58~61
value	object ID 2	object coord x	object coord y	current time in utc	reserved

Note: If "object num in this packet" is 0, then all data has been sent, and connection will be closed.

9.4.5 Subscribe Heat Map Raw Data

Table 9-46

URL	http://<server>/cgi-bin/HeatMapManager.cgi?action=attachRaw					
Method	GET					
Description	Subscribe the heap map raw data.					
[Request Params] (key=value format in URL)						
Name	Type	R/O	Param Description			
channel	int	O	Video channel index which starts from 1, default is 1.			
heartbeat	int	O	Send heartbeat interval, range is [1, 60],unit is second. If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client, the heartbeat message are "Heartbeat". If this parameter is not present, its default value is 60.			
[Response Params] (multipart binary data)						
[Example]						
Request	GET http://192.168.1.250/cgi-bin/HeatMapManager.cgi?action=attachRaw&channel=1&heartbeat=5					
Response	HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<boundary> Connection: closed --<boundary> Content-Type: application/octet-stream Content-Length: <length> <heap map raw data> --<boundary> Content-Type: text/plain Content-Length: 11 Heartbeat -<boundary> Content-Type: application/octet-stream Content-Length: <length>					

	<pre><heap map raw data> --<boundary> ...</pre>
--	---

- **Appendix A:** The heap map raw data format.

octet	0	1	2	3	4 ~ 23
value	width		height		StartTime, ex: "2012-01-04 00:00:00"
octet	24~43			44 ~ 47	
value	EndTime, ex: "2012-01-04 01:00:00"			reserved	
octet	48	49	...		
value	Data: every byte symbolize a pixel				

9.5 Crowd Distribute Map

9.5.1 Get Channel Caps

Table 9-47

Syntax	http://<server>/cgi-bin/crowdDistrMap.cgi?action=getCaps
Method	GET
Description	Get Channel CrowdDistriMap Caps.
Example	http://192.168.1.250/cgi-bin/crowdDistrMap.cgi?action=getCaps
Success Return	CrowdCapsList[0].channel=< ChannelNo > CrowdCapsList[0].Support=< Support > CrowdCapsList[1].channel=< ChannelNo > CrowdCapsList[1].Support=< Support > ...
Comment	Parameters in URL and Response: ChannelNo : video channel index Support : true or false, support or not

9.5.2 Subscribe to Realtime Crowd Stat

Table 9-48

Syntax	http://<server>/cgi-bin/crowdDistrMap.cgi?action=attach&channel=< ChannelNo >[&heartbeat=< Heartbeat >]
Method	GET
Description	Subscribe the crowd distribute map information, return info at regular time.
Example	http://<server>/cgi-bin/crowdDistrMap.cgi?action=attach&channel=1&heartbeat=5
Success Return	HTTP/1.1 200 OK Content-Type: multipart/x-mixed-replace; boundary=<boundary> Connection: close

	<pre>--<boundary> Content-Type: text/plain Content-Length: <data length> CrowdStatData[0].Channel=1 CrowdStatData[0].GloabalPeopleNum =10 CrowdStatData[0].RegionNum =1 CrowdStatData[0].RegionPeopleList[0].RegionID=0 CrowdStatData[0].RegionPeopleList[0].Region[0][0]=10 CrowdStatData[0].RegionPeopleList[0].Region[0][1]=10 CrowdStatData[0].RegionPeopleList[0].Region[1][0]=10 CrowdStatData[0].RegionPeopleList[0].Region[1][1]=100 ... CrowdStatData[0].RegionPeopleList[0].RegionPeopleNum=100 CrowdStatData[0].CrowdEventNum =2 CrowdStatData[0].CrowdList[0].Center=2 CrowdStatData[0].CrowdList[0].Radius=2 CrowdStatData[0].RegionEventNum =2 CrowdStatData[0].RegionList[0].Region[0][0]=10 CrowdStatData[0].RegionList[0].Region[0][1]=10 CrowdStatData[0].RegionList[0].Region[1][0]=10 CrowdStatData[0].RegionList[0].Region[1][0]=100 ... CrowdStatData[0].RegionList[0].RegionID=0 CrowdStatData[0].RegionList[0].PeopleNum=100 --<boundary> Content-Type: text/plain Content-Length: 11 Heartbeat --<boundary> Content-Type: text/plain Content-Length: <data length> CrowdStatData[0].Channel=1 ...</pre>
Comment	<p>Parameters in URL and Response</p> <p>ChannelNo: integer, video channel index to subscribe</p> <p>Region is Polygon, has a list of points, every point has x and y , so Region[0][0] is first point's x, Region[0][1] is first point's y, Region[1][0] is second point's x, Region[1][1] is second point's y, and so on</p> <p>Heartbeat: integer, range is [1,60],unit is second.If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client,the heartbeat meaage are "Heartbeat".</p>

9.5.3 Get Current Crowd Stat

Table 9-49

Syntax	<code>http://<server>/cgi-bin/crowdDistrimap.cgi?action=getSummary&channel=<ChannelNo></code>
Method	GET
Description	Get crowd distribute map information, return info only once.
Example	<code>http://<server>/cgi-bin/crowdDistrimap.cgi?action=getSummary&channel=1</code>
Success Return	<pre> CrowdStatData[0].Channel=1 CrowdStatData[0].GloabalPeopleNum =10 CrowdStatData[0].RegionNum =1 CrowdStatData[0].RegionPeopleList[0].RegionID=0 CrowdStatData[0].RegionPeopleList[0].Region[0][0]=10 CrowdStatData[0].RegionPeopleList[0].Region[0][1]=10 CrowdStatData[0].RegionPeopleList[0].Region[1][0]=10 CrowdStatData[0].RegionPeopleList[0].Region[1][1]=100 ... CrowdStatData[0].RegionPeopleList[0].RegionPeopleNum=100 CrowdStatData[0].CrowdEventNum =2 CrowdStatData[0].CrowdList[0].Center=2 CrowdStatData[0].CrowdList[0].Radius=2 CrowdStatData[0].RegionEventNum =2 CrowdStatData[0].RegionList[0].Region[0][0]=10 CrowdStatData[0].RegionList[0].Region[0][1]=10 CrowdStatData[0].RegionList[0].Region[1][0]=10 CrowdStatData[0].RegionList[0].Region[1][0]=100 ... CrowdStatData[0].RegionList[0].RegionID=0 CrowdStatData[0].RegionList[0].PeopleNum=100 </pre>
Comment	<p>Similar to above "Subscribe to Realtime Crowd Stat", but reply stat only once, not at regular time.</p> <p>ChannelNo: integer, video channel index which starts from 1.</p>

9.6 Video Analyse

9.6.1 Get Video Analyse Capability

Table 9-50

Syntax	<code>http://<server>/cgi-bin/devVideoAnalyse.cgi?action=getcaps&channel=<ChannelNo></code>
Method	GET
Description	Get video analyse capabilities.
Example	<code>http://192.168.1.250/cgi-bin/devVideoAnalyse.cgi?action=getcaps&channel=1</code>
Success Return	<pre> caps.CalibrateBoxs[0]=2 caps.CalibrateBoxs[1]=3 </pre>

	<pre> caps.ComplexSizeFilter=false caps.MaxCelibateAreas=10 caps.MaxExcludeRegions=0 caps.MaxInternalOptions=512 caps.MaxModules=1 caps.SupportGlobalDeviceParam=1 caps.MaxPointOfLine=20 caps.MaxPointOfRegion=20 caps.MaxRules=10 caps.MaxStaffs=4 caps.SpecifiedObjectFilter=true caps.SupportedRules[0]=CrossLineDetection caps.SupportedRules[1]=CrossRegionDetection caps.SupportedRules[2]=LeftDetection caps.SupportedRules[3]=TakenAwayDetection caps.SupportedScene[0]=Normal caps.SupportedScene[1]=FaceDetection caps.SupportedScene[2]=VideoDiagnosis caps.SupportedScenes.FaceDetection.SupportedCalibrateParams.Groud.HorizontalStaffs[0]=0 caps.SupportedScenes.FaceDetection.SupportedCalibrateParams.Groud.HorizontalStaffs[1]=0 caps.SupportedScenes.FaceDetection.SupportedCalibrateParams.Groud.VerticalStaffs[0]=0 caps.SupportedScenes.FaceDetection.SupportedCalibrateParams.Groud.VerticalStaffs[1]=0 caps.SupportedScenes.StereoNumber.SupportedRules.ManNumDetection.SupportLocalDataStore=false caps.SupportedScenes.NumberStat.CameraType=1 caps.SupportedScenes.NumberStat.SupportedRules.NumberStat.MaxRules=8 caps.SupportedScenes.StereoBehavior.OnlyFindLastEvtList[0]="ManStandDetection" </pre>
Comment	<p>Parameters in URL: ChannelNo: integer, video channel index which starts from 1.</p>

Appendix

ParamName	ParamValue type	Description
caps.SupportedScenes.StereoBehavior.OnlyFindLastEvtList	array<string>	event type list in which last ones are supported to be inquired.

9.6.2 [Config] Video Analyse Global

- Get video analyse global config

Table 9-51

Syntax	<a href="http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=VideoAnalyseGlob">http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=VideoAnalyseGlob
--------	---

	al
Method	GET
Description	Get video analyse global config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=VideoAnalyseGlobal
Success Return	<pre> head.Scene.Type=Normal head.Scene.PtzPresetId=1 head.Scene.Depth=Far head.Scene.Detail.CameraAngle=30 head.Scene.Detail.CameraDistance=10.000000 head.Scene.Detail.CameraHeight=6.200000 head.TimePeriod.Day[0]=8:00:00 head.TimePeriod.Day[1]=20:00:00 head.TimePeriod.Night[0]=20:00:00 head.TimePeriod.Night[1]=8:00:00 ... </pre>
Comment	<p>Parameters in Response :</p> <p>head =table.VideoAnalyseGlobal[ChannelNo]</p> <p>ChannelNo: integer, array index starts from 0, which means video channel>equals to video channel index -1, and so 0 means channel 1).</p>

- Set video analyse global config

Table 9-52

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set video analyse global config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoAnalyseGlobal[0].Scene.Type=Normal&VideoAnalyseGlobal[0].Scene.PtzPresetId=1
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>paramName and paramValue are as table below.</p> <p>In table below, head =VideoAnalyseGlobal[ChannelNo]</p> <p>ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p> <p>ParamName start with head.Scene.Detail depends on head.Scene.Type.</p>

Appendix

ParamName	ParamValue type	Description
head .Scene.Type	string	Scene class, the range is { "", "Normal", "Indoor", "ATM", "Traffic", "FaceRecognition", "FaceDetection", "NumberStat", "HeatMap", "VideoDiagnosis", "VehicleAnalyse", "TrafficPatrol", "CourseRecord", "Vehicle" , "ObjectDetect", "VehicleCompare"}

ParamName	ParamValue type	Description
head.Scene.PtzPresetId	integer	Range is 0—255, 0 means that the scene is unassociated with PTZ.
head.Scene.Depth	string	Picture distance feature, the range is { "Normal", "Far", "Middle", "Near" }
head.Scene.Detail.Value	string	Detail config of a scene. For example, when Scene.Type is "Normal", it's detail includes CameraAngle, CameraDistance, CameraHeight, etc.
head.TimePeriod.Day[0]	string	The start time of Day, it's format is hh:mm:ss
head.TimePeriod.Day[1]	string	The end time of Day
head.TimePeriod.Night[0]	string	The start time of Night, it's format is hh:mm:ss
head.TimePeriod.Night[1]	string	The end time of Night

9.6.3 [Config] Video Analyse Rule

- Get video analyse rule

Table 9-53

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=VideoAnalyseRule
Method	GET
Description	Get video analyse rules config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=VideoAnalyseRule
Success Return	head.Name = line1 head.Type =CrossLineDetection head.VideoAnalyseRule[0][0].Enable =true head.VideoAnalyseRule[0][0].EventHandler = (output of EventHandler is described in GetEventHandler) ...
Comment	Parameters in Response : head =table.VideoAnalyseRule[ChannelNo] [RuleNo] ChannelNo : integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). RuleNo =rule index.

- Set video analyse rule

Table 9-54

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue> [&<paramName>=<paramValue> ...]
Method	GET
Description	Set video analyse rules config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&VideoAnalyseRule[0][0].Name=myAnalyseRule1&VideoAnalyseRule[0][0].Type=CrossLineDetection
Success	OK

Return	
Comment	<p>Parameters in URL: paramName and paramValue are as table below.</p> <p>In table below, head=VideoAnalyseRule[ChannelNo] [RuleNo]</p> <p>ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p> <p>RuleNo =rule index.</p> <p>ParamName start with head.Config is only effective with {"CrossLineDetection", "CrossRegionDetection", "LeftDetection", "TakenAwayDetection"}.</p>

Appendix

ParamName	ParamValue type	Description
head.Name	string	Rule name, it must be unique.
head.Type	string	The range is {"CrossLineDetection", "CrossRegionDetection", "LeftDetection", "TakenAwayDetection", "VideoAbnormalDetection", "FaceDetection", "AudioMutation", "AudioAnomaly", "VideoUnFocus", "WanderDetection", "RioterDetection", "ParkingDetection", "MoveDetection", "NumberStat", "HeatMap", "FaceAttribute", "VehicleDetect", "NonMotorDetect", "HumanTrait", "ObjectPlacement", "ObjectRemoval" }.
head.Enable	bool	Enable/Disable this rule.
head.EventHandler	—	Setting of EventHandler is described in SetEventHandler .
head.Config.DetectLine[0][0]	integer	The start point of DetectLine 0;
head.Config.DetectLine[0][1]	integer	The end point of DetectLine 0;
head.Config.DetectLine[1][0]	integer	The start point of DetectLine 1;
head.Config.DetectLine[1][1]	integer	The end point of DetectLine 1;
head.Config.Direction	string	The range is {"LeftToRight", "RightToLeft", "Both"}.
head.Config.SizeFilter.MaxSize[0]	integer	Maximum width. The width of the object must not be beyond maximum width. Adapt to {"CrossLineDetection", "CrossRegionDetection", "LeftDetection", "TakenAwayDetection", "FaceDetection", "WanderDetection", "RioterDetection", "ParkingDetection", "MoveDetection"}.
head.Config.SizeFilter.MaxSize[1]	integer	Maximum height. The height of the object must not be beyond maximum height.
head.Config.SizeFilter.MinSize[0]	integer	Minimum width. The width of the object must not be less than minimum width.
head.Config.SizeFilter.MinSize[1]	integer	Minimum height. The height of the object must not be beyond minimum height.

ParamName	ParamValue type	Description
head.Config.DetectRegion[0][0]	integer	The start point of DetectRegion 0; Adapt to {"CrossRegionDetection", "LeftDetection", "TakenAwayDetection", "WanderDetection", "RioterDetection", "ParkingDetection", "MoveDetection"}.
head.Config.DetectRegion[0][1]	integer	The end point of DetectRegion 0;
head.Config.DetectRegion[1][0]	integer	The start point of DetectRegion 1;
head.Config.DetectRegion[1][1]	integer	The end point of DetectRegion 1;
head.Config.DetectRegion[2][0]	integer	The start point of DetectRegion 2;
head.Config.DetectRegion[2][1]	integer	The start point of DetectRegion 2;
head.Config.MinDuration	integer	Range is 1—600, adapt to {"LeftDetection", "TakenAwayDetection", "WanderDetection"}. Range is 10-300, adapt to {"RioterDetection"}. Range is 6-300, adapt to {"ParkingDetection"}.
head.Config.Sensitivity	integer	Range is 1—10, adapt to {"RioterDetection", "MoveDetection"}.
head.Config.EnterThreshold	integer	Range is 0—100000000, adapt to {"NumberStat"}.
head.Config.ExitThreshold	integer	Range is 0—100000000, adapt to {"NumberStat"}.
head.Config.InsideThreshold	integer	Range is 0—100000000, adapt to {"NumberStat"}.
head.Config.DirectionStats.Enable	bool	Enable/Disable objectdetect Direction. Rule range is {"VehicleDetect", "NonMotorDetect", "HumanTrait"}.
head.Config.DirectionStats.DetectLine[0][0]	integer	The start point of DetectLine 0, Rule range is {"VehicleDetect", "NonMotorDetect", "HumanTrait"}.
head.Config.DirectionStats.DetectLine[0][1]	integer	The start point of DetectLine 0, Rule range is {"VehicleDetect", "NonMotorDetect", "HumanTrait"}.
head.Config.DirectionStats.DetectLine[1][0]	integer	The start point of DetectLine 1, Rule range is {"VehicleDetect", "NonMotorDetect", "HumanTrait"}.
head.Config.DirectionStats.DetectLine[1][1]	integer	The start point of DetectLine 1, Rule range is {"VehicleDetect", "NonMotorDetect", "HumanTrait"}.

ParamName	ParamValue type	Description
		"HumanTrait".
head .Config.DirectionStats.Direction	string	The range is {"LeftToRight", "RightToLeft", "Both"}. Rule range is {"VehicleDetect", "NonMotorDetect", "HumanTrait"}.
head .Config.FilterUnAliveEnable	bool	Enable the non-living filtering function. It is false by default.
head .Config.EyesDistThreshold	uint32	Set filtering threshold of pupillary distance. If the actual pupillary distance is less than the threshold, it will be filtered. Refer to GB/T 35678-2017: Value range [50–no upper limit] required for face recognition application image technology. The default value 0 indicates no filtering.
head .Config.HelmetEnable	bool	Enable safety helmet detection. With the function enabled, the face recognition result carries information related to safety helmet detection. It is false by default.
head .Config.TempSwitch	uint8	Select temperature monitoring status: 0: Disable temperature monitoring 1: Normal temperature monitoring 2: Debug temperature monitoring
head .Config.TempModel	uint8	Temperature monitoring mode (valid when TempSwitch is not 0): 0: Automatical mode 1: Thermal image detection mode 2: Calibration mode
head .Config.TempStrategy.TempValueMax	double	Maximum value of normal face temperature range
head .Config.TempStrategy.TempValueMin	double	Minimum value of normal face temperature range
head .Config.TempStrategy.TempType	uint8	Strategy type for calculating face temperature: 0: Use the highest temperature 1: Use the average temperature
head .Config.TempStrategy.SlideNum	uint8	The number of sliding cache frames. 0 means no sliding, and the maximum cache is 32 frames.
head .Config.TempStrategy.HighTempStrategy.TempDetectRegion	uint8	Parameter for maximum temperature strategy (valid when tempType is 0). Calculate the highest temperature area: 0: Forehead

ParamName	ParamValue type	Description
head .Config.TempStrategy.AverageTempStrategy.StrategyType	uint8	1: Whole face Parameter for maximum temperature strategy (valid when tempType is 1). Average temperature strategy type: 0: Center point 1: high temperature point
head .Config.TempStrategy.AverageTempStrategy.PointNum	uint8	Parameter for average temperature strategy (valid when tempType is 1). The number of points used to find the average value, and it is the square of the integer, such as 4, 9, and 16.
head .Config.TempStrategy.AverageTempStrategy.EnableFilter	bool	Parameter for average temperature strategy (valid when tempType is 1). Whether to enable the removal of the highest temperature and the lowest temperature to take the average value: False: Disable True: Enable
head .Config.FilterMaskUnAliveEnable	bool	Select whether to enable liveness detection for person wearing mask. This value is related to FilterUnAliveEnable. When FilterUnAliveEnable is true (liveness detection is enabled), this field is valid.

Get Last Event Info

Get Last Event Info

Table 9-55

Syntax	http://<server>/cgi-bin/devVideoAnalyse.cgi?action=getLastEventInfo&channel=< ChannelNo >&ClassName=< ClassNameStr >&EventName=< EventNameStr >
Method	GET
Description	Get last event info
Example	http://192.168.1.250/cgi-bin/devVideoAnalyse.cgi?action=getLastEventInfo&channel=1&ClassName=StereoBehavior&EventName=ManStandDetection
Success Return	State=1 EventInfo.Code=ManStandDetection EventInfo.Action=Start
Comment	Parameters in Response : ChannelNo : integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ClassNameStr : str, means Class name. EventNameStr : str, means Event name.

9.6.5 [Config] GlobalDeviceParam

- Get Global Device Param

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=GlobalDeviceParam		
Method	GET		
Description	Get Global Device Param		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
table	object	R	
+LocateHeight	Float	R	Device locate Height
[Example]			
Request	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=GlobalDeviceParam		
Response	table.GlobalDeviceParam.LocateHeight=10.5		

- Set Global Device Param

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]		
Method	GET		
Description	Set Global Device Param		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
GlobalDeviceParam	object	R	Global Device Param
+LocateHeight	float	R	Device locate Height
[Response Params] (OK at Body)			
Name	Type	R/O	Param Description
[Example]			
Request	http://<server>/cgi-bin/configManager.cgi?action=setConfig&GlobalDeviceParam.LocateHeight = 10.5		
Response	Ok		

9.6.6 Get Template Rule

Gets all intelligent rule configuration templates and default values under the specified category

Request URL	http://<server>/cgi-bin/VideoInAnalyse.cgi?action=getTemplateRule		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Class	Char[32]	R	the same meaning with Type in config VideoAnalyseGlobal
			Example
			"Normal"

			"Normal" "Traffic" "TrafficPatrol" "FaceDetection"(shared by face detect and face recognition) in added, Web can use "ALL" for all rules.	
Channel	integer	R	video channel, which starts from 1.	1
Example				
<code>http://<server>/cgi-bin/VideoInAnalyse.cgi?action=getTemplateRule&Channel=1&Class=Normal</code>				

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
Rule	object	R	rule info	
+Normal	object	R	business class	
++CrossRegionDetection	object	R	cross region templet	
++CrossLineDetection	object	R	default config of crossline format as the same as each object in the second dimension of VideoAnalyseRule	
+++Config	object	R	detail of rule config	
++++SizeFilter	object	R		
+++++MinSize	int[2]	R		[0, 0]
+++++MaxSize	int[2]	R		[8191, 8191]
+++++DetectLine	object[]	R		
+++++Direction	char[]	R		"Both"
+++PtzPresetId	integer	R	Preset number	0
+++ObjectType	char[][]	R	type of detect object	["Unknown"]
+++Type	char[64]	R	rule type	"CrossLineDetection"
+++Class	char[16]	R	business plan class	"Normal"
+++Enable	bool	R	enable the rule	false
+++Id	integer	R	rule id	0

Example

```

Rule.Normal.CrossLineDetection.Id=0
Rule.Normal.CrossLineDetection.Enable=false
Rule.Normal.CrossLineDetection.Class=Normal
Rule.Normal.CrossLineDetection.Type=CrossLineDetection
Rule.Normal.CrossLineDetection.ObjectTypes[0]=Unknown
Rule.Normal.CrossLineDetection.PtzPresetId=0
Rule.Normal.Config.Direction=Both
Rule.Normal.Config.SizeFilter.MaxValue[0]=[8191, 8191]
Rule.Normal.Config.SizeFilter.MinValue[0]=[0, 0]

```

...

9.6.7 [Config] IntelliSchemeTourEnableSetting

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&IntelliSchemeTour.Enable=<enableValue>			
Method	SET			
Request Params	key=value format at URL			
Name	Type	R/O	Description	Example
IntelliSchemeTour	object	O	IntelliSchemeTour Enable Setting	
+Enable	bool	R	IntelliSchemeTour Enable Setting	True
Example	http://<server>/cgi-bin/configManager.cgi?action=setConfig&IntelliSchemeTour.Enable=true			

Request Params	key=value format at URL			
Name	Type	R/O	Description	Example
Example				
OK				

9.6.8 [Config] Intelligent Tour Plan

Get intelligent tour plan

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=IntelliSchemeTour			
Method	GET			
Parameter Format	key=value format at URL			
Parameter	Type	Required	Description	Example
IntelliSchemeTour	object	Yes	Intelligent package tour plan	
Example	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=IntelliSchemeTour			

Parameter Format	key=value format at URL			
Parameter	Type	Required	Description	Example
IntelliSchemeTour	object	No	Intelligent package tour plan. No channel sensitive.	
+Enable	bool	No	Whether the package tour plan is enabled.	true

+TourPriMode	enumchar[32]	No	<p>Tour priority mode</p> <p>enumchar[32]{</p> <p>"Switch": Scene priority, tour in strict accordance with the defined tour time.</p> <p>"Detect": Detection priority, with the complete set of data as the condition for scene switching upon completion of scene detection (For illegal parking detection cameras. Multiple images may be needed in one scene. In this case, the camera can rotate to the next scene after all images are captured).</p> <p>"Tour": Tour priority, similar to detection priority, with a set of data as the condition for scene switching upon completion of scene detection (both close shot and long shot available. Apply close shot first and then long shot according to the tour path).</p> <p>}</p> <p>For details, refer to the capability set IntelliScheme.getCa</p>	"Switch"
--------------	--------------	----	--	----------

			ps.	
+IdleWaitingTime	uint32	No	Idle waiting time, that is, the idle delay time until the tour plan continues to take effect after a user stops operating the device, in second(s).	10
+SceneSwitchingTime	uint32	No	Scene switching time. When the tour plan is enabled and the priority mode is either detection priority or tour priority, it will automatically rotate to the next scene after an object cannot be detected for a certain period of time.	20
+TourPlan	object[7][16]	No	Package tour plan 2D array. The first dimension indicates the day of a week, totally 7 weekdays, and the second dimension indicates the tour groups supported that day.	
++TimeSection	TimeSection	No	Period of time. 00:00:00 indicates the start time and 23:59:59 indicates the end time.	"1 00:00:00-23:59:59"
++TourPath	object[]	No	Tour path 1D array, to adjust the order of packages. Up to 10 packages are supported. It is null by default and added by a user. Field values are	null

			specified by a user.	
+++SchemeID	uint32	No	Intelligent package ID, value range: 1-300.	1
+++Duration	uint32	No	Retention time, value range: 30-36,000 s or 0. It is 0 by default if this field is unavailable. Indicates that the switching has not been made during the period of time.	600
+++Speed	uint32	No	The speed of the camera rotating to the preset when a package includes a preset channel of a PTZ camera, value range: 1-10 speed levels.	7

Example

```
table.IntelliSchemeTour.Enable=true
table.IntelliSchemeTour.IdleWaitingTime=10
table.IntelliSchemeTour.SceneSwitchingTime=20
table.IntelliSchemeTour.TourPlan[0][0].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[0][1].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[0][2].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[0][3].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[0][4].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[0][5].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[0][6].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[0][7].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[0][8].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[0][9].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[0][10].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[0][11].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[0][12].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[0][13].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[0][14].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[0][15].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[1][0].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[1][1].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[1][2].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[1][3].TimeSection=1 00:00:00-00:00:00
```


9.6.9 Export Intelligent Diagnosis, Allowlist, and Blocklist Information

Export information, such as intelligent diagnosis, allowlist, and blocklist information. The export data format is csv

Request URL	http://<server>/cgi-bin/api/lmExport/exportData			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
type	enumint	O	Exported data type. enumint{ 0: License plate allowlist 1: License plate blocklist 2: License plate blocklist and allowlist 3: Voice talk contacts 4: Custom password 5: Card recordset 6: Face information for access control 7: Fingerprints for access control 8: IPC information of voice talk 9: Calling person information 10: Public password recordset 11: Operation and maintenance information }	0
Request Example				
{ "type": 1 }				

Response Params (binary in body)				
Name	Type	R/O	Description	Example
Response Example				
HTTP/1.1 200 OK Server: Device/1.0 Content-Type: application/octet-stream Content-Length: <length> < binary data >				

9.6.10 Import Intelligent O&M, Allowlist, and Blocklist Information

Import intelligent O&M, allowlist, and blocklist information.

Request URL	http://<server>/cgi-bin/api/lmExport/importData
--------------------	---

Method	POST			
Request Params (multipart in body)				
Name	Type	R/O	Description	Example
type	uint32	R	Imported data type enumint{ 0 : License plate allowlist 1 : License plate blocklist 2 : License plate blocklist and allowlist 3: Voice talk contacts 4: Custom password 5: Card recordset 6: Face information for access control 7: Fingerprint for access control 8: IPC information of voice talk 9: Calling person information 10: Public password recordset 11: Operation and maintenance information }	0
isOverWrite	bool	R	Whether to overwrite the original data. If isOverWrite=true, clear all license plate lists, and then import the data. If isOverWrite=false, it will be imported directly, even if the license plate to be imported is duplicated with the existing license plate of the device.	true
Request Example				
POST http://<server>/cgi-bin/api/lmExport/importData HTTP/1.1 User-Agent: client/1.0 Content-Type: multipart/x-mixed-replace; boundary=<boundary> --<boundary> Content-Type: application/json Content-Length: 12 { "type": 0, "isOverWrite": true } --<boundary> Content-Type: application/octet-stream Content-Length: 800 <i>< binary data></i>				

--<boundary>

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

9.6.11 Get Intelligent Capability

URL	http://<server>/cgi-bin/intelli.cgi?action=getCaps		
Method	GET		
Description	Get video analyse intelligent capabilities.		
[Request Params] (key=value)			
capsName	string	O	Name of capability, it can be: AnalyseMode: get analyse mode capacity MultiChannelMode: get multi channel seperate intelligent capacity and combined intelligent capacity MultiVideoAnalyse: get all intelligent capacity from the multichannel device in each channel if omit, means get all capacity
[Response Params] (key=value)			
caps	object	R	The video service capabilities.
+AnalyseMode	string	O	Analyse mode, can be : RealStream, RecordFileStream, PicFileStream
+MultiChannelMode	object	O	Multi channel description
++IndepMode	Array<Arra y<object>>	O	Multi channel open at one time separately
+++Channel	int	R	Video channel number, start from 0
+++Type	string	R	Business class
++CompMode	Array<Arra y<object>>	O	Multi channel combined mode
+++Channel	int	R	Video channel number, start from 0
+++Type	string	R	Business class
+MultiVideoAnalyse	Array<obje ct>	O	List of channel capabilities
++caps	object	R	Video analyse capabilities
+Algorithm	Array<obje ct>	O	Algorithm version information
++Class	string	R	Intelligent analyse class, ex : ObjectDetect
++AlgorithmVersion	string	R	Algorithm version
++AlgorithmVendor	sting	R	Algorithm Vendor name
+TotalCapacity	Array<obje ct>	O	Intelligent analyse total capacity

++Class	string	R	Intelligent analyse class, ex : ObjectDetect
++Type	Array<string>	R	Intelligent analyse rules, ex : FaceDetection
++Number	int	R	The maximum number of video channels that can be analysed at same time
[Example]			
Request	GET http://192.168.1.250/cgi-bin/intelli.cgi?action=getCaps caps.AnalyseMode=RealStream caps.MultiChannelMode.IndepMode[0][0].Channel=0 caps.MultiChannelMode.IndepMode[0][0].Type=Normal caps.MultiChannelMode.IndepMode[0][1].Channel=1 caps.MultiChannelMode.IndepMode[0][1].Type=ObjectDetect ... caps.MultiChannelMode.CompMode[0][0].Channel=0 caps.MultiChannelMode.CompMode[0][0].Type=Normal caps.MultiChannelMode.CompMode[0][1].Channel=1 caps.MultiChannelMode.CompMode[0][1].Type=ObjectDetect ... caps.Algorithm[0].Class=Normal caps.Algorithm[0].AlgorithmVersion=V2.8 caps.Algorithm[0].AlgorithmVendor=XXX caps.Algorithm[1].Class=ObjectDetect caps.Algorithm[1].AlgorithmVersion=V2.8 caps.Algorithm[1].AlgorithmVendor=YYY ... caps.TotalCapacity[0].Class=Normal caps.TotalCapacity[0].Type[0]=FaceDetection caps.TotalCapacity[0].Number=3 caps.TotalCapacity[1].Class=ObjectDetect caps.TotalCapacity[1].Type[0]=FaceDetection caps.TotalCapacity[1].Number=3 ...		
Response			

9.6.12 Subscribe Resource Usage Info

URL	http://<server>/cgi-bin/intelli.cgi?action=attachResource		
Method	GET		
Description	Subscribe the intelligent analyse resource		
[Request Params] (key=value format in URL)			
heartbeat	int	O	Send heartbeat interval, range is [1, 60], unit is second. If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client, the heartbeat message are "Heartbeat". If this parameter is not present, its default value is 60.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description

RemainCapacity	Array<object>	R	Remain capacity of intelligent analyse
+Class	string	R	Intelligent analyse class, ex : ObjectDetect
+Number	int	O	Remain number of video channels that can be analysed
[Example]			
Request	GET http://192.168.1.250/cgi-bin/intelli.cgi?action=attachResource&heartbeat=5		
Response	HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<boundary> Connection: closed --<boundary> Content-Type: text/plain Content-Length: <length> RemainCapacity[0].Class=Normal RemainCapacity[0].Number=1 RemainCapacity[1].Class=ObjectDetect RemainCapacity[1].Number=2		
	--<boundary> Content-Type: text/plain Content-Length: 11 Heartbeat --<boundary> Content-Type: text/plain Content-Length: <length> RemainCapacity[0].Class=Normal RemainCapacity[0].Number=1 RemainCapacity[1].Class=ObjectDetect RemainCapacity[1].Number=1 ...		

9.6.13 Export Encrypted Files

Request URL	http://<server>/cgi-bin/api/SecurityImExport/exportData			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
key	char[1024]	NA	Password string encrypted with public key and encoded by Base64	"1BBzEVMS***"
type	enumint	NA	Exported types enumint{ 0: License plate allowlist	0

		1: License plate blocklist 2: License plate allowlist/blocklist 3: Voice talk contacts 4: Custom password 5: Card recordset 6: Face information for access control 7: Fingerprint for access control 8: Network camera information of voice talk 9: Calling person information 10: Public password recordset 11: Operation and maintenance information 12: Face data 13: Face images }	
--	--	---	--

Request Example

```
{
  "key": "1BBzEVMS***",
  "type": 0
}
```

Response Params (binary in body)				
Name	Type	R/O	Description	Example
Response Example				
HTTP/1.1 200 OK				
Server: Device/1.0				
Content-Type: application/octet-stream				
Content-Length: <length>				
< binary data>				
//Before the device returns the actual data, use AES symmetric encryption (the key is generated according to the plaintext conversion after key decryption).				
//AES-256bit-CBC. Filling mode: PKCS7.				

9.6.14 Platform intelligent control

Request URL	http://<server>/cgi-bin/api/intelli/setPollingConfig			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Config	object[]	O	Length max 256	
+Enable	bool	O	Whether it is intelligently turned on	true

+Channel	uint32	O	Channel number	12
+GlobalType	char[64]	O	Global configuration, only support single channel and single intelligence,	"Normal"
+RuleType	char[16][64]	O	Rule configuration, which represents all rules enabled in the GlobalType or GlobalTypeList scenario	["CrossLineDetection"]
+GlobalTypeList	char[6][64]	O	The global configuration list, the extension supports the opening of multiple intelligence of a single channel, choose one of GlobalType and GlobalTypeList, GlobalTypeList will be used first	["Normal", "GasStation"]

Request Example

```
{
  "Config": [
    {
      "Enable": true,
      "Channel": 12,
      "GlobalType": "Normal",
      "RuleType": ["CrossLineDetection"],
      "GlobalTypeList": ["Normal", "GasStation"]
    }, ...
  ]
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

9.6.15 Get Scene List

Get the list of intelligent schemes that have been opened for a channel.

Request URL	http://<server>/cgi-bin/devVideoAnalyse.cgi?action=getSceneList			
Method	GET			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Request Example				
http://<server>/cgi-bin/devVideoAnalyse.cgi?action=getSceneList				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Sample
VideoAnalyseGlobal	object[]	R	List of intelligent schemes opened for each video channel.	
+Scenes	char[32][16]	R	If none of the intelligent schemes are enabled, return VideoAnalyseGlobal[0].Scenes[0]=""	["Normal", "HeatMap"]
Response Example				
VideoAnalyseGlobal[0].Scenes[0]=Normal VideoAnalyseGlobal[0].Scenes[1]=FaceDetection				

9.6.16 Open Intelligent Schemes

Open several intelligent schemes for a channel

Request URL	http://<server>/cgi-bin/devVideoAnalyse.cgi?action=enableScene&typeList=<TypeList>&channel=<ChannelNo>			
Method	GET			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
type	char[32][16]	R	For the list of intelligent schemes to open, refer to getSceneList	["Normal", "HeatMap"]
channel	int32	R	Video channel, counted from 0	0
Request Example				
http://192.168.1.250/cgi-bin/devVideoAnalyse.cgi?action=enableScene&typeList=[Normal,HeatMap]&channel=0				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Sample
response	char[256]	R	Result "OK" is returned for success, and "Error" is returned for failure	"OK"
Response Example				
OK				

1.1.1 Close Intelligent Schemes

Close several intelligent schemes for a channel

Request URL	http://<server>/cgi-bin/devVideoAnalyse.cgi?action=disableScene&typeList=<TypeList>&channel=<ChannelNo>			
Method	GET			

Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
typeList	char[32][16]	R	For the list of intelligent schemes to close, refer to getSceneList	["Normal", "HeatMap"]
channel	int32	O	Video channel, counted from 0	0
Request Example				
http://192.168.1.250/cgi-bin/devVideoAnalyse.cgi?action=disableScene&typeList=[Normal,Heat Map]&channel=0				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Sample
response	char[256]	R	Result "OK" is returned for success, and "Error" is returned for failure	"OK"
Response Example				
OK				

9.7 WorkSuitCompareServer

9.7.1 Add Compliance Library

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/createGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Group	object	R	Compliance library group info	
+GroupName	char[128]	R	Compliance library group name, up to 127 characters.	"takeout"
+CutoutPolicy	uint32	O	Preferred solution 0: Full body 1: Upper body	1
+Similarity	uint8	O	Similarity threshold, range [1, 100], default 67	90
+GroupType	enumchar[16]	O	Compliance library group type enumchar[16]{ "BlockListDB": Registry (control list) to save external imported data }	"BlockListDB"
+GroupDetail	char[256]	O	group remark information	"community owner"
+Type	enumchar[16]	O	type, default: "WorkSuit" enumchar[16]{	"ClothesCommon"

		<pre>"WorkSuit" "ClothesCommon" } In the future, there may be traffic police comparison, takeout comparison and worker comparison, and "clothescommon" may be used for comparison through clothing characteristics</pre>	
--	--	---	--

Request Example

```
{
  "Group": {
    "GroupName": "takeout",
    "CutoutPolicy": 1,
    "Similarity": 90,
    "GroupType": "BlockListDB",
    "GroupDetail": "community owner",
    "Type": "ClothesCommon"
  }
}
```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
GroupID	char[64]	R	Compliance library group ID	"10000"
Response Example				
<pre>{ "GroupID": "10000" }</pre>				

9.7.2 Delete Compliance Library

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/deleteGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
GroupID	char[64]	R	Compliance library group ID	"000001"
Request Example				
<pre>{ "GroupID": "000001" }</pre>				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				

{}

9.7.3 Find Compliance Library

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/findGroup		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
GroupID	char[64]	R	Compliance library group ID, "" for all Compliance library group IDs
Request Example			
<pre>{ "GroupID": "000001" }</pre>			

Response Params (JSON format in body)			
Name	Type	R/O	Description
GroupList	object[64]	O	List of found compliance library group information
+GroupID	char[64]	O	Compliance library group ID
+GroupName	char[128]	O	Compliance library group name, up to 127 characters.
+CutoutPolicy	uint32	O	Preferred solution 0: Full body 1: Upper body
+GroupType	enumchar[16]	O	Compliance library group type enumchar[16]{ "BlockListDB": Registry (control list) to save external imported data }
+GroupDetail	char[256]	O	group remark info
+GroupSize	int	O	当前组内工装数
+Channels	int16[1024]	O	The list of video channel numbers to which the current group is bound (see setgroup). If not associated with any video channel, it should be [-1]
+Similarity	uint8[1024]	O	It corresponds to the similarity threshold when Binding video. The array length is equal to channels. It is meaningless when no video channel is bound. Each value range [1, 100]
+FeatureState	uint[4]	O	The number of work suit in various states in the group and the work suit
			[10,20,30,40]

		<p>that have not completed modeling (feature extraction) cannot be identified by the algorithm</p> <p>Array subscript correspondence [0] - the number of work suit ready for modeling does not guarantee the success of modeling</p> <p>[1] - the number of work suit failed in modeling, the picture does not meet the algorithm requirements, and the picture needs to be replaced</p> <p>[2] - the number of work suit successfully modeled, and the data can be used for work suit identification by algorithm</p> <p>[3] - the number of models that have been successfully modeled but become unavailable due to algorithm upgrade, it will be available after remodeled.</p>	
--	--	---	--

Response Example

```
{
  "GroupList": [
    {
      "GroupID": "000001",
      "GroupName": "taken",
      "CutoutPolicy": 0,
      "GroupType": "BlockListDB",
      "GroupDetail": "community owner",
      "GroupSize": 30,
      "Channels": [0],
      "Similarity": [90],
      "FeatureState": [10,20,30,40]
    },...{}
  }
}
```

9.7.4 Get Compliance Library Arming Information of Channels

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/getGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Channel	int	R	Video channel number	0
Request Example				
{	<pre> "Channel": 0 }</pre>			

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
GroupList	object[128]	O	List of compliance library group information	
+GroupID	char[64]	O	Compliance group ID, up to 63 characters.	"001"
+Similarity	uint8	O	Similarity threshold; value range: 1–100.	80
+CutoutPolicy	uint32	O	Preferred solution 0: Full body 1: Upper body	1

Response Example

```
{
  "GroupList": [
    {
      "GroupID": "001",
      "Similarity": 80,
      "CutoutPolicy": 1
    }, ...
  ]
}
```

9.7.5 Modify Compliance Group Information

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/modifyGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Group	object	R	Compliance library group info	
+GroupID	char[64]	R	Compliance group ID, up to 63 characters	"0001"
+GroupName	char[128]	O	Compliance library group name, up to 127 characters.	"park"
+Similarity	uint8	O	Similarity threshold, range [1, 100], default 67	90
+GroupType	enumchar[16]	O	Compliance library group type enumchar[16]{ "BlockListDB": Registry (control list) to save external imported data }	"BlockListDB"
+GroupDetail	char[256]	O	group remark info	"community owner"
Request Example				
<pre>{ "Group": { "GroupID": "0001", "GroupName": "park", "Similarity": 90, "GroupType": "BlockListDB", "GroupDetail": "community owner" } }</pre>				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				{}

9.7.6 Deploy Compliance Library

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/setGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Channel	int	R	video channel number	0
GroupList	object[128]	O	Arming information list: If the list does not exist, the arming of all compliance libraries for the channels are deleted.	
+GroupID	char[64]	O	Compliance group ID, up to 63 characters	"001"
+Similarity	uint8	O	Similarity threshold for each compliance group. Action after the comparison similarity is higher than the threshold Absence of this field means no modification Similarity range [0, 100]	80
Request Example				
<pre>{ "Channel": 0, "GroupList": [{ "GroupID": "001", "Similarity": 80 }, ...] }</pre>				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				{}

9.7.7 Find Workwear Information in Compliance Library

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/startFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Condition	object	O	Search condition.	
+GroupID	char[128][64]	O	Array list of compliance library IDs. Each compliance library ID is up to 63 characters.	["00001","00002"]
WorkSuit	object	O	Search condition of workwear	

			information	
+FeatureState	uint8	O	Feature value status: 0: Unknown; 1: Extraction failed; 2: Extracted successfully; 3: Modeled but algorithm upgrade causes data unavailability and remodeling is required.	1

Request Example

```
{
  "Condition": {
    "GroupID": ["00001","00002"]
  },
  "WorkSuit": {
    "FeatureState": 1
  }
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Token	uint	否	Token for search task, which is used to get search results and stop the search.	2342343
TotalCount	int	否	Total number of results returned.	3333

Response Example

```
{
  "Token": 2342343,
  "TotalCount": 3333
}
```

9.7.8 Get Find Workwear Information Result

Acquire the work uniform information from the search result.

Note: If doFind operation is not performed for 60 seconds, the returned token will be invalid.

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/doFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Condition	object	R	Search for the input parameter of work uniform.	
+Token	uint	R	Search token.	46878
+BeginNumber	uint	R	Start number of the search. The search starts from the “beginNumber” records, and returns	0

			predefined number (count) of records. 0<=beginNumber<= totalCount-1	
+Count	int	R	Number of entries obtained each time. It should not exceed 50.	20
+NeedData	uint	O	Image format mask returned by search results. It is added for IPC. bit0 means to return the HTTP link of image. For example, 1 only returns HTTP link. If the field does not exist, the server decides how to return it.	1
Request Example				
<pre>{ "Condition": { "Token": 46878, "BeginNumber": 0, "Count": 20, "NeedData": 1 } }</pre>				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Results	object	R	Output parameter of the search.	0
+Found	int	R	The actual number of entries returned by this search.	1
+Candidates	object[]	R	Information list of work uniform to be selected. Maximum 50 information lists can be returned at one time.	
++WorkSuit	object	R	Searched work uniform information list.	12
+++UID	string	O	Work uniform identifier. The maximum length is 31 characters.	0005
+++GroupID	string	O	Compliance database ID. The maximum	10000

			length is 63 characters.	
+++GroupName	string	O	Name of work uniform group.	Group1
+++FeatureState	uint	O	Feature status. 0: Unknown. 1: Extraction failed. 2: Extraction succeeded. 3: Algorithm upgrade makes data unavailable and it needs to be remodeled. 4: Calculating features.	0
+++FeatureErrCode	uint	O	Recorded reasons for modeling failure. Only valid when FailedINDt is greater than 0. 0: Unknown (other errors) 1: Modeling failed. 2: System errors (for example, errors caused by invalid license and unstated modeling analyzer). 3: Database operation failed. 4: image decoding failed. 5: Multiple targets. 6: No target.	0
+++ImagePath	string	O	Work uniform thumbnail path.	/mnt/2010/8/1/dav/15:40:50.jpg
+++SourceUID	string	O	Unique identifier of panoramic image. It is generated by the platform.	1
+++SourceFileName	string	O	Panoramic image name.	aaa
+++Image	object	O	Image information	[,]
++++ Width	uint	O	Image width	100
++++ Height	uint	O	Image height	50
++++ Offset	uint	O	Offset in the binary data block	0
++++ Length	uint	O	Image size	10000
Response Example				

```
{
  "Results": {
    "Found": 12,
    "Candidates": [
      {
        "WorkSuit": [
          {
            "UID": "111",
            "GroupID": "0001",
            "GroupName": "group1",
            "FeatureState": 1,
            "FeatureErrCode": 2,
            "ImagePath": "/mnt/2010/8/1/dav/15:40:50.jpg",
            "SourceUID": "1",
            "SourceFileName": "aaaaa",
            "Image": {
              "Width": 1920,
              "Height": 1080,
              "Offset": 0,
              "Length": 1156
            }
          }, ...
        ]
      }
    ]
  }
}
```

9.7.9 Stop Find Workwear Information

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/stopFind		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
Token	uint	R	Token for search task, which is used to get search results and stop the search.
Request Example			
{	"Token": 46878		
}			

Response Params (JSON format in body)
Name
Type
R/O
Description
Example
Response Example
{}

9.7.10 Delete Workwear Information

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/deleteByUID		
Method	POST		
Request Params (JSON format in body)			

Name	Type	R/O	Description	Example
UID	char[64][32]	O	Workwear identifier list, up to 31 characters.	["123", "234"]
Request Example				
{ "UID": ["123", "234"] }				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
ErrCode	uint[64]	O	Error code list, indicating error messages when deleting information. 0: Success. 1: The workwear does not exist. 2: Database operation failed.	[0,0]
Response Example				
{ "ErrCode": [0,0] }				

9.7.11 Re-extracting Features by Workwear

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/reAbstract			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
WorkSuit	object[64]	O	Multiple work suit information, [] means to reconstruct all work suit with missing or mismatched feature vectors	
+UID	char[32]	O	Workwear identifier list; each UID is up to 31 characters.	"120837"
Token	uint	O	The token value of the work clothes modeling. The token can be obtained through getappendtoken	1
Request Example				
{ "WorkSuit": ["UID": "120837"], "Token": 1 }				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				

{}

9.7.12 Stop Re-extracting Workwear Features

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/stopReAbstract		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
Request Example			
{}			

Response Params (JSON format in body)			
Name	Type	R/O	Description
Response Example			
{}			

9.8 Smart Motion Detection

9.8.1 [Config] SmartMotionDetect

- Get smart motion detection setting

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=SmartMotionDete ct					
Method	GET					
Description	Get smart motion detection setting					
[Request Params] (None)						
[Response Params] (key=value format)						
Name	Type	R/O	Param Description			
table	object	R	Config info table.			
+SmartMotionDetec t	Array<object>	R	PrivacyMasking config, each channel has one config object			
++Enable	bool	R	Enable/Disable			
++Sensitivity	string	R	Detection sensitivity, can be: "Low", "Middle", "High"			
++ObjectTypes	object	R	Detection object type			
+++Human	bool	R	Whether detect motion of human			
+++Vehicle	bool	R	Whether detect motion of vehicle			
[Example]						
Request	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=SmartMotionDe tect					
Response	table.SmartMotionDetect[0].Enable=true table.SmartMotionDetect[0].Sensitivity=Middle table.SmartMotionDetect[0].ObjectTypes.Human=true table.SmartMotionDetect[0].ObjectTypes.Vehicle=true ...					

- Set smart motion detection setting

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set smart motion detection setting		
[Request Params] (key=value format)			
Name	Type	R/O	Param Description
SmartMotionDetect	Array<object>	R	Smart motion detection config, each channel has one config object
+Enable	bool	R	Enable/Disable
+Sensitivity	string	R	Detection sensitivity, can be: "Low", "Middle", "High"
+ObjectTypes	object	R	Detection object type
++Human	bool	R	Whether detect motion of human
++Vehicle	bool	R	Whether detect motion of vehicle
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&SmartMotionDetect[0].Enable=true&SmartMotionDetect[0].Sensitivity=Middle&SmartMotionDetect[0].ObjectTypes.Human=true&SmartMotionDetect[0].ObjectTypes.Vehicle=true		
Response	OK		

9.8.2 Start SMD Data Search

Request URL	http://<server>/cgi-bin/api/SmdDataFinder/startFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Condition	object	R	Search conditions	
+Channel	int32	O	Channel number. -1 means searching for all channels	0
+Channels	int[]	O	Channel number array (starting from 0). There should be at least one field, Channel or Channels. Generally, If both Channel and Channels fields exist, Channels should prevail.	
+SmdType	char[8][32]	O	Video type. Array "smdTypeHuman": Human. "smdTypeVehicle": Vehicle. "smdTypeHumanAndVehicle": Human and vehicle.	["smdTypeHuman", "smdTypeVehicle", "smdTypeHumanAndVehicle"]
+StartTime	char[32]	O	Start time	"2017-08-01 00:00:00"
+EndTime	char[32]	O	End time	"2017-08-02 00:00:00"
+Order	char[16]	O	Search for ordering method. "ascOrder": Ascending order (default value).	"ascOrder"

			"descOrder": Descending order.	
Request Example				
{				
"Condition": {				
"Channel": 0,				
"Channels": [0,1,2,5],				
"SmdType": ["smdTypeHuman", "smdTypeVehicle", "smdTypeHumanAndVehicle"],				
"StartTime": "2017-08-01 00:00:00",				
"EndTime": "2017-08-02 00:00:00",				
"Order": "ascOrder"				
}				
}				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Token	int	O	Return searched ID	1
Count	int	O	Total number of searched entries	100
Response Example				
{				
"Token": 1,				
"Count": 100				
}				

9.8.3 Get SMD Data Search Result

Request URL	http://<server>/cgi-bin/api/SmdDataFinder/doFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Token	int	R	Search ID	1
Offset	int32	R	Location offset	0
Count	int	R	Number of searched items	100
Request Example				
{				
"Token": 1,				
"Offset": 0,				
"Count": 100				
}				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
SmdInfo	object[]	R	It is an array. Each element represents the historical data of a record data at a certain time.	
+Channel	int32	O	Channel number, starting from 0	1

+EndTime	char[32]	O	End time. "201	"2017-08-02 00:00:00"
+Type	char[32]	O	Video type. "smdTypeHuman": Human. "smdTypeVehicle": Vehicle. "smdTypeHumanAndVehicle": Human and vehicle.	"smdTypeHuman"
+StartTime	char[32]	O	Start time	2017-01-01 10:00:

Response Example

```
{
  "SmdInfo": [
    {
      "Channel": 1,
      "StartTime": "2017-08-01 00:00:00",
      "EndTime": "2017-08-02 00:00:00",
      "Type": "smdTypeHuman"
    }, ...
  ]
}
```

9.8.4 End SMD Data Search

Request URL	http://<server>/cgi-bin/api/SmdDataFinder/stopFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Token	int	O	Search ID	1
Request Example				
{ "Token": 1 }				

Response Params (JSON format in body)
Name
Type
R/O
Description
Example
Response Example
{}

9.9 Intelligent analysis tasks

9.9.1 Add task

Request URL	http://<server>/cgi-bin/api/analyseTaskManager/add			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Task	object	R	Task object	
+TaskUserData	char[512]	R	Task data, add it intact	"asdvtgrffg"

a			to the response	
+SourceType	char[32]	R	<p>Data source type</p> <p>Use devVideoAnalyse component for local streams (LocalStream), and it is not included in this component group.</p> <p>"LocalStream": Local stream</p> <p>Currently, the local stream is used only by NVR and SMD. Use devVideoAnalyse for other cases.</p> <p>"RemoteStream": Remote real stream</p> <p>"LocalVideoFile": Local video files</p> <p>"LocalPictureFile": Local picture files</p> <p>"RemoteVideoFile": Remote video files</p> <p>"RemotePictureFile": Remote picture files</p> <p>"PushVideoFile": Actively pushed video files</p> <p>"PushPictureFile": Actively pushed picture files</p> <p>"OffLineVideoFile": Offline video files (from third parties)</p> <p>PushPictureFileByRule: Remote picture files.</p> <p>Rules and picture information are not included when adding tasks. Through pushing the picture interface, each picture is attached with different rule information (Used in energy-related scenes currently).</p> <p>"IOTData": IoT device data</p>	"StreamSource"

			For active pushed files such as PushVideoFile, the command does not specify the file information, but pushes data through command such as analysePushVideoFile.	
+IsStart	bool	O	Starts immediately after being created or not (Valid when the file type is LocalVideoFile, LocalPictureFile, RemoteVideoFile, and RemotePictureFile. Add, attach and then start in case of result loss) If the value is true, the command will start immediately after being created.	true
+LocalStream	object	O	Local real-time video sources Valid when SouceType is LocalStream	
++Channel	int	O	Video channel	0
++Subtype	uint8	O	Stream type: 0: Main stream 1: Sub stream 1 2: Sub stream 2	0
+RemoteStrea m	object	O	Remote real-time stream source Valid when SouceType is RemoteStream	
++Protocol	char[32]	O	Video stream protocol Supported protocols by now: "Private" "RTSP" "Onvif" "GB28181" "HIKVISION" "BSCP"	"RTSP"
++Path	char[256]	O	Video stream URL	"rtsp://10.35.114.145:554/EVMS/monitor/param?camera=AGfQv8TwA1ALIPO29LA9L2&substream=0"

				m=1&trackID=801"
++IP	char[64]	O	IP Address	"10.35.114.145"
++Port	uint16	O	Port	554
++UserName	char[64]	O	Username	"admin"
++Password	char[64]	O	Password	"admin123"
++Channel	int	O	Video channel	0
++Subtype	uint8	O	Stream type: 0: Main stream 1: Sub stream 1 2: Sub stream 2	0
++ChannelId	char[256]	O	Platform channel No, used to bind the device channels.	"1000254\$1\$0\$110"
+LocalVideoFil e	object	O	Local video file, and only 1 video file is supported Valid when SouceType is LocalVideoFile	
++FileID	char[128]	O	File ID	"file-1234"
++Path	char[260]	O	File name	"/xxxx/a.dav"
+LocalPicture File	object	O	Local picture file, and multiple picture files are supported. Valid when SouceType is LocalPictureFile	
++Files	object[]	O		
+++FileID	char[128]	O	File ID	"file-1234"
+++Path	char[260]	O	File name	"/xxxx/a.jpg"
+RemoteVide oFile	object	O	Remote video file, and only 1 video file is supported. Valid when SouceType is RemoteVideoFile	
++Protocol	char[16]	O	Protocol for accessing remote files "RTSP", "RabbitMq" "Private" "Onvif" "GB28181" "HIKVISION" "BSCP" (FTP and RTSP are expected to be covered)	"RTSP"
++VideoAnaly sisProcess	uint32	O	Video analysis progress. When SourceType is RemoteVideoFile, the valid value range is 0–	66

			100. 100 means the analysis is compete.	
++Path	char[260]	O	File path	"rtsp://192.168.1.1:554/xxx/a.dav"
++IP	char[64]	O	Remote IP address	"192.168.1.1"
++Port	int	O	Remote port	554
++Channel	int	O	Video channel	1
++Subtype	uint8	O	Stream type: 0: Main stream 1: Sub stream 1 2: Sub stream 2	1
++Username	char[64]	O	Username	"admin"
++Password	char[64]	O	Password	"123456"
++FileID	char[128]	O	File ID	"file-1234"
++StartTime	char[64]	O	Start time	"2010-05-25 00:00:00"
++EndTime	char[64]	O	End time	"2010-05-25 23:59:59"
+OffLineVideoFile	object	O	Remote video file, and only 1 video file is supported (the file imported from a third party). Valid when SouceType is OffLineVideoFile	
++Protocol	char[16]	R	Protocol for accessing remote files RTSP	"RTSP"
++FileSize	uint64	O	File size (in bytes)	1000
++VideoAnalysisProcess	uint32	O	Video analysis progress. When SourceType is OffLineVideoFile, the valid value range is 0–100. 100 means the analysis is compete.	66
++Path	char[260]	R	File path	"rtsp://192.168.1.1:554/xxx/a.dav"
++IP	char[64]	R	Remote IP address	"192.168.1.1"
++Port	int	R	Remote port	4000
++Username	char[64]	R	Username	"admin"
++Password	char[64]	R	Password	"123456"
++FileID	char[128]	R	File ID	"file-1234"
++isReportPTS	bool	O	Reports the relative timestamp or not	true
+RemotePictureFile	object	O	Remote picture file, and multiple picture files are supported. The server must be identical.	

			Valid when SouceType is RemotePictureFile	
++Protocol	char[16]	O	Protocol for accessing remote files HTTP, RabbitMq, ActiveMq (FTP and RTSP are expected to be covered)	"HTTP"
++IP	char[64]	O	Remote IP address	"192.168.1.1"
++Port	int	O	Remote port	80
++Username	char[64]	O	Username	"admin"
++Password	char[64]	O	Password	"123456"
++Files	object[]	O	File list	
+++FileID	char[128]	O	File ID	"file-1234"
+++Path	char[260]	O	File name	"http://192.168.1.1:80/xx xx/a.jpg"
+Global	object	O	Global settings	
++Lanes	object[8]	O	Lane information; each lane includes 2 border lines. Maximum length: 8 arrays For highway all-in-one device, a channel supports up to 8 pairs of lane lines.	
+++Enable	bool	O	Enable lane The rules take effect only when the lane is enabled. It tackles the detection errors when a lane is under construction in a specific period. When the filed is empty, the lane is enabled by default.	true
+++Number	int	O	Lane No.	0
+++LeftLine	uint16[20][2]	O	Left lane line. The direction of the lane line refers to the lane direction, and the line at the left side of the lane is left lane line. The coordinates of the point is unified within the range of 0–8192.	

+++LeftLineExt	uint16[64][2]	O	LeftLineExt is the extension filed of LeftLine. When the point number of the left lane line is less than 20, use LeftLine only. When the point number of the left lane line is more than 20, use LeftLine for the former 20 points, and LeftLineExt for the rest. LeftLine and LeftLineExt comprise the left lane line.	
+++LeftLineType	enumchar[16]	O	Attributes of the left lane line enumchar[16]{ "WhiteSolid": White solid line "WhiteDotted": White dotted line "Yellow": Yellow line "UpSolidDownDotted": Solid at the top and dotted at the bottom "UpDottedDownSolid": Dotted at the top and solid at the bottom	"WhiteSolid"
+++RightLine	uint16[20][2]	O	Right lane line. The direction of the lane line refers to the lane direction, and the line at the right side of the lane is right lane line. The coordinates of the point is unified within the range of 0–8192.	
+++RightLineExt	uint16[64][2]	O	RightLineExt is the extension filed of RightLine.	++++RightLineExt
+++RightLineType	char[16]	O	Attributes of the right lane line Refer to LeftLineType	"WhiteSolid"
++CalibrateArea	object[]	O	Calibration Area An array. Each scene includes multiple calibration areas. If the	

			<p>file does not exist, the whole scene is the calibration area.</p> <p>CalibrateArea, CalibrateArea1</p> <p>CalibrateAreaN refers to scene, scene 1..... scene N.</p> <p>For highway all-in-one device, a channel supports 1 calibration area.</p>	
+++Area	uint[20][2]	O	<p>Calibrates a polygon area, and uses all.</p> <p>For highway all-in-one device: 1 area.</p>	
+++Staffs	object[]	O	<p>Ruler line</p> <p>A calibration area requires multiple ruler lines.</p> <p>Use 4 elements for depth of field calibration (3 vertical elements and 1 horizontal element)</p> <p>Use 2 elements for lane calibration (an element for each side)</p> <p>No element is needed for license plate calibration (illegal parking), and use VideoAnalyseCalibrate to configure separately.</p> <p>No element is needed for dual-PTZ calibration, and draw an area.</p> <p>LaneNew only covers 1 vertical straight line segment, and the coordinates of the vertical straight line must be entered.</p> <p>For highway all-in-one device, 1 Staffs for a calibration area.</p>	
++++Type	enumchar[32]	O	<p>Ruler type</p> <p>enumchar[32]{</p>	"Horizontal"

			<p>"Horizontal": Horizontal line segment "Vertical": Vertical line segment "Any": Any line segment, unused "Cross": Cross line segment, unused }</p> <p>For highway all-in-one device (1 staffs): 1 type</p>	
++++Start	int[2]	O	<p>Coordinates of the start point The coordinates is unified within the range of 0–8192</p>	[0, 0]
++++End	int[2]	O	<p>Coordinates of the end point The coordinates is unified within the range of 0–8192</p>	[100, 100]
++++Length	double	O	<p>Actual length, in meters Note: The current system layer length, in meters</p>	1.0
++GlobalDetectionEnable	bool	O	Enable detection identifiers in the area	false
++ObjectArea	object[20]	O	Arrays in the target detection area	
+++Area	uint16[20][2]	O	Area, refer to the configurations of DetectRegion.	
+++Type	enumchar[16]	O	<p>Target area type enumchar[16]{ "SignalLight": Signal light "LEDScreen": LED screen }</p>	"SignalLight"
++CameraHeight	float	O	Installation height of the camera in meters	6.2
+Module	object	O	Module configuration	
++ExcludeRegion	uint16[][][20][2]	O	Zones that should be excluded from the detection area, usually zones that might	[, ...,]

			<p>interfere with the algorithms. A detection area might have 0 or more excluded zones.</p> <p>Array of areas. For the definition of area, see DetectRegion.</p> <p>The maximum number depends on the capability set.</p> <p>For highway all-in-one device, up to 10 excluded areas are supported.</p>	
++SizeFilter	SizeFilter	O	<p>Physical size filter by default.</p> <p>If both the default object type and a specific filter are selected, choose the specific filter.</p> <p>Highway all-in-one device supports a maximum box and a minimum box.</p>	SizeFilter
++ObjectFilter	object[]	O	The specific filter for each object. It is a number.	
+++ObjectType	char[16]	O	Object type. For available values, see ObjectTypeEnum.	"Vechicle"
+++SizeFilter	SizeFilter	O	Size filter	+++SizeFilter
+++Enable	bool	O	Enable the size filter of the specific object or not	true
++DetectRegion	uint16[20][2]	O	Detection area. Coordinates of the peaks of the polygon.	+++DetectRegion
++SpecialDetectRegion	object[10]	O	Specifies the general service detection area and exclusion area. Up to 10 service types are supported.	
+++Type	char[16]	O	Which service scheme category does the detection area parameter suits. For available values, see VAClassEnum.	"Normal"

+++DetectRegion	uint16[10][20][2]	O	Detection area The coordinates of the peaks is unified within the range of 0–8192.	
+++ExcludeRegion	uint16[10][20][2]	O	Zones that should be excluded from the detection area, usually zones that might interfere with the algorithms. A detection area might have 0 or more excluded zones.	[, ...,]
+Rules	object[]	R	Analysis rule	
++Class	char[16]	R	Analysis category The values are identical with the service scheme categories of VideoAnalyseGlobal.	"ObjectDetect"
++Type	char[32]	R	Analysis rule type The values are identical with the rule types of VideoAnalyseRule.	"ObjectDetect"
++Name	char[128]	R	Rule name. Rule name of devices without presets must be different. For devices with presets, the rule name in the same preset must be different. The name of rules in different presets can overlap.	"GuardLine1"
++ObjectType s	char[16][16]	O	List of detected object types Note: Not all rules need the object type. It is used primarily for the earliest behavior algorithms. We recommend you add the parameter to rule config. Here it is not changed due to compatibility. It is identical with VideoAnalyseRule(Base). When only the object	["Human", "Vehicle"]

			<p>type in the array is detected, the array cannot be empty. See VideoAnalysModule.ObjectType</p> <p>When ObjectTypes is empty, or there is unknown filed in the array, no matter whether there is other element, it means the classifier is not enabled.</p> <p>If Unknown does not exist in the array, only specific object types will be detected.</p>	
++Config	object	R	For specific rules, see "VideoAnalyseRule(XXX X)	
++Experience Config	object[128]	O	Experience database configuration array	
+++ID	char[64]	O	Experience database ID	"1"
+++Similarity	uint8	O	Similarity threshold: The range is [1–100], and it is 67 by default.	67
+++IsPositive	bool	O	True or not: true: True false: False	true
+MQConfig	char[4096]	O	MQ configurations. See PaaS protocol configuration center > Operator configuration. When the remote access type is RabbitMq or ActiveMq, try to acquire MQ from this field. ActiveMq is used for the exchange of sensitive information.	""
+IsRepeat	int	O	Whether repetition is allowed. 0 means R, and it is the default value. 1 means no.	0

Request Example

```
{
  "Task": {
```

```

    "TaskUserData": "asdvbtgrffg",
    "SourceType": "StreamSource",
    "IsStart": true,
    "LocalStream": {
        "Channel": 0,
        "Subtype": 0
    },
    "RemoteStream": {
        "Protocol": "RTSP",
        "Path": "rtsp://10.35.114.145:554/EVMS/monitor/param?cameraid=AGfQv8TwA1ALIPO29LA9L2&substream=1&trackID=801",
        "IP": "10.35.114.145",
        "Port": 554,
        "UserName": "admin",
        "Password": "admin123",
        "Channel": 0,
        "Subtype": 0,
        "ChannelId": "1000254$1$0$110"
    },
    "LocalVideoFile": {
        "FileID": "file-1234",
        "Path": "/xxxx/a.dav"
    },
    "LocalPictureFile": {
        "Files": [
            {
                "FileID": "file-1234",
                "Path": "/xxxx/a.jpg"
            }, ...
        ]
    },
    "RemoteVideoFile": {
        "Protocol": "RTSP",
        "VideoAnalysisProcess": 66,
        "Path": "rtsp://192.168.1.1:554/xxxx/a.dav",
        "IP": "192.168.1.1",
        "Port": 554,
        "Channel": 1,
        "Subtype": 1,
        "Username": "admin",
        "Password": "123456",
        "FileID": "file-1234",
        "StartTime": "2010-05-25 00:00:00",
        "EndTime": "2010-05-25 23:59:59"
    },
    "OffLineVideoFile": {
        "Protocol": "RTSP",
        "FileSize": 1000,

```

```
"VideoAnalysisProcess": 66,
"Path": "rtsp://192.168.1.1:554/xxxx/a.dav",
"IP": "192.168.1.1",
"Port": 4000,
"Username": "admin",
"Password": "123456",
"FileID": "file-1234",
"isReportPTS": true
},
"RemotePictureFile": {
"Protocol": "HTTP",
"IP": "192.168.1.1",
"Port": 80,
"Username": "admin",
"Password": "123456",
"Files": [
{
"FileID": "file-1234",
"Path": "http://192.168.1.1:80/xxxx/a.jpg"
},...{}]
},
"Global": {
"LANes": [
{
"Enable": true,
"Number": 0,
"LeftLine": ,
"LeftLineExt": ,
"LeftLineType": "WhiteSolid",
"RightLine": ,
"RightLineExt": +++++RightLineExt,
"RightLineType": "WhiteSolid"
},...{}],
"CalibrateArea": [
{
"Area": ,
"Staffs": [
{
>Type": "Horizontal",
"Start": [0, 0],
"End": [100, 100],
"Length": 1.0
},...{}]
},...{}],
"GlobalDetectionEnable": false,
"ObjectArea": [
{
"Area": ,
>Type": "SignalLight"
},...{}],
"CameraHeight": 6.2
},
```

```

"Module": {
    "ExcludeRegion": [, ..., ],
    "SizeFilter": SizeFilter,
    "ObjectFilter": [
        "ObjectType": "Vechicle",
        "SizeFilter": ++++SizeFilter,
        "Enable": true
    ],...{}],
    "DetectRegion": ++++DetectRegion,
    "SpecialDetectRegion": [{{
        "Type": "Normal",
        "DetectRegion": ,
        "ExcludeRegion": [, ..., ]
    }},...{}]
},
"Rules": [
    {
        "Class": "ObjectDetect",
        "Type": "ObjectDetect",
        "Name": "GuardLine1",
        "ObjectTypes": ["Human", "Vehicle"],
        "Config": {
        },
        "ExperienceConfig": [{{
            "ID": "1",
            "Similarity": 67,
            "IsPositive": true
        }},...{}]
    },...{}],
    "MQConfig": "",
    "IsRepeat": 0
}
}

```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
TaskID	uint32	Non e	Task ID, > 0. (If it is failed, the result is false)	1234
VirtualChannel	uint32	Non e	The corresponding virtual channel number of the task. The value range is within the values of ChannelManager.getVirtualChannels.	1234
Path	char[256]	O	RTSP address of the AI stream	"rtsp://192.168.1.1:554/x"

Response Example

```
{

```

```

    "TaskID": 1234,
    "VirtualChannel": 1234,
    "Path": "rtsp://192.168.1.1:554/xx"
}

```

9.9.2 Attach task result

Subscribe to all results (do not specify TaskID), or subscribe to specific TaskID. If you want to subscribe to the results of newly created tasks, to avoid event loss, we recommend you not set immediate start when creating the tasks. You can subscribe to the results first, and then run the start command.

Request URL	http://<server>/cgi-bin/api/analyseTaskManager/attachResult		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
TaskID	uint[64]	O	Subscribed TaskID When it is not entered, or the array is empty, the results of all tasks are subscribed to.
Filters	object	O	Filter conditions for subscription
+Events	char[64][32]	O	Filtered event
+ImageDataFlag	uint32	O	0 means containing pictures. It is the default value. 1 means not containing pictures.
+ImageDataType	char[16][16]	O	Optional. It is used to combine the images of the event. One-dimensional array. If "ImageData-Type" does not exist, it means reporting all the pictures of the event. If there is an element in the array, it reports pictures based on the element requirements. The value of each element: 1. "ObjectImage": Reporting target cutout 2. "SceneImage": Reporting scene image
Heartbeat	int	O	Server keep-alive; integer; Unit: s;
			5

		The value range is [1–60]. For example, if the parameter exists in the URL and the value is 5, the device will send “heartbeat” to the client as a keep-alive message. Note: The keep-alive message must be sent before the keepalive parameter expires.	
--	--	--	--

Request Example

```
{
    "TaskID": [ 1, 2, 3 ],
    "Filters": {
        "Events": [ "FaceDetection", ... ],
        "ImageDataFlag": 0,
        "ImageDataType": ["ObjectImage", "ScenelImage"],
        "Heartbeat": 5
    }
}
```

Response Params (JSON format in body)				
Name	Name	Name	Name	Name
SID	uint32	R	Subscribed SID	1234U
Infos	object[]	None		
+TaskID	uint	None	Task ID	1
+TaskUserData	char[512]	None	Task data	"asdvtgrffg"
+UserData	char[64]	O	Video source data, including the video source information, and it corresponds to the UserData field in addPollingTask.	"asdvtgrffg"
+TaskCustomData	TaskCustomData	O	Custom data	
+UserDefineData	char[512]	O	Custom data by the user, and it corresponds to the UserDefineData field in analyseTaskManager.a	

			analysePushPictureFileByRule.	
+FileID	char[128]	O	File ID. Valid when analyzing the file, and offered when a file analysis task is complete.	"file-1234"
+FileState	int	O	File analysis status: 0: Analyzing 1: Complete 2: Failed	0
+FileAnalyseMsg	char[256]	O	Analyzing additional information, usually the failure reason.	"Decode Failed, Download Failed"
+Events	object[]	O	Event information, such as TrafficJunction, HumanTrait and more. For the format, see EventInfo in EventManager. Offered when the analysis results come out. Simultaneously reports up to 32 events.	[@{Cplusplus InformationExchange Format-Management Branch.doc#EventInfo}Info]//end of Events
+Flag	uint32	O	Tag status: 1: Unprocessed 2: Valid 3: Invalid	2
+Image	object	O	Linked snapshots	
++Offset	uint	None	Offset of images in binary data	0
++Length	uint	None	Image length (in bytes)	60000
Response Example				
HTTP/1.1 200 OK				
Content-Type: multipart/x-mixed-replace; boundary=<boundary>				
Connection: close				
--<boundary>				
Content-Type: application/json				

```

Content-Length: <data length>

{
    "SID": "1234U",
    "Infos": [
        {
            "TaskID": 1,
            "TaskUserData": "asdvtgrffg",
            "UserData": "asdvtgrffg",
            "TaskCustomData": null,
            "FileID": "file-1234",
            "FileState": 0,
            "FileAnalyseMsg": "Decode Failed, Download Failed",
            "Events": [{}],
            "Flag": 2,
            "Image": {
                "Offset": 0,
                "Length": 60000
            }
        }
    ]
}

--<boundary>
Content-Type: text/plain
Content-Length: 11
Heartbeat
--<boundary>
...

```

9.9.3 Remove task

Request URL	http://<server>/cgi-bin/api/analyseTaskManager/remove			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
TaskID	uint32	None	Task ID	1
Request Example				
{				

```

        "TaskID": 1
    }
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

9.9.4 Push Picture File

Valid when data source type is PushPictureFile. support pushing multiple files at the same time, each file should be completed, not support chunk

Request URL	http://<server>/cgi-bin/api/analyseTaskManager/analysePushPictureFile			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
TaskID	uint	是	Task ID	1
Files	object[]	是	File list URL path and binary data are supported, but only one way can be selected at a time	
+FileID	char[128]	是	File ID	"file-1234"
+Url	char[512]	否	Remote file URL , with the necessary information for access, including username, password	"ftp://username:password@hostname:port/filepath"
+Offset	uint	否	Offset in binary data, bytes	0
+Length	uint	否	File size, bytes	256000
+XRayCustomInfo	XRayCustomInfo	否	Customer information, X-ray machine customization	
+ModelPath	char[512]	否	Algorithm model path	"/aaa/bbb/ccc"

Request Example

```
{
    "TaskID": 1,
    "Files": [
        {
            "FileID": "file-1234",
            "Url": "ftp://username:password@hostname:port/filepath",
            "Offset": 0,
            "Length": 256000,
            "XRayCustomInfo": ,
            "ModelPath": "/aaa/bbb/ccc"
        }, ...
    ]
}
```

```
}
```

--<boundary>

Content-Type: image/jpeg

Content-Length:<image size>

<JPEG image data>

--<boundary>

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

10.Intelligent Traffic API

10.1 Intelligent Traffic Event

10.1.1 [Event] TrafficJunction

Table 10-1

Usage	Refer to "4.4.3 Subscribe to Snapshot"for how to subscribe event		
Description	When detects vehicle passing, send this event.		
[Event Params] (key=value format)			
Name	Type	R/O	Param Description
Events	array<object>	R	Event info array.
+EventBaseInfo	object	R	Base info of event.
++Code	String	R	Event Code. It should be TrafficJunction .
++Action	String	R	Event Action. It can be "Start", "Stop" or "Pulse".
++Index	int	O	The channel index relate to this event.
+GroupID	int	O	The id of event group.
+CountInGroup	int	O	Event count in the event group.
+IndexInGroup	int	O	The index of this event in the event group, start from 1.
+Lane	int	O	Lane number, start from 0.
+TriggerType	int	O	Trigger type. It can be: 0—car detector, 1—radar, 2—video.
+Speed	int	O	Vehicle speed, unit is km/hour.
+Vehicle	Object	O	The information of vehicle object.
++BoundingBox	Array<int>	R	The detected car bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
++Text	string	O	Vehicle logo.
++SubText	string	O	Vehicle sub logo.
++SubBrand	int	O	Vehicle sub brand index.
++BrandYear	int	O	Vehicle brand year index.
+TrafficCar	object	O	Traffic Car info.
++ReINDo	int	R	The record id.
++PlateNumber	string	R	Car plate number.
++PlateType	string	O	Plate type.
++PlateColor	string	O	Plate color, ex: "Yellow", "Blue", ... etc.
++VehicleColor	string	O	Vehicle color, ex: "Yellow", "Blue", ... etc.
++BoundingBox	Array<int>	R	The detected plate bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point.

			Coordinate remap to 0 — 8192.
++Country	string	O	Country info. Max string length is 19.
++Speed	int	O	Vehicle speed, unit is km/hour.
++Event	string	O	The event info, ex: "TrafficJunction".
+CommlInfo	object	O	Traffic event common info.
++Seat	array<object>	O	Vehicle front seat info.
+++Type	string	O	Front seat type. It can be: "Main""Slave".
+++Status	array<string>	O	Some driver status. It can be some of the following: "Smoking", "Calling".
+++SunShade	string	O	Sunshade status. It can be: "Unknow", "WithSunShade", "WithoutSunShade".
+++ShadePos	Array<int>	O	The detected sun shade bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
+++SafeBelt	string	O	SafeBelt status. It can be: "Unknow", "WithSafeBelt", "WithoutSafeBelt".

[Example]

Event	--<boundary> Content-Type: text/plain Content-Length: <length> Events[0].EventBaseInfo.Code=TrafficJunction Events[0].EventBaseInfo.Action=Pulse Events[0].EventBaseInfo.Index=0 Events[0].GroupID=123 Events[0].CountInGroup=3 Events[0].IndexInGroup=1 Events[0].Lane=0 Events[0].Vehicle.BoundingBox[0]=1341 Events[0].Vehicle.BoundingBox[1]=2451 Events[0].Vehicle.BoundingBox[2]=4513 Events[0].Vehicle.BoundingBox[3]=4135 Events[0].Vehicle.Text=Audi Events[0].Vehicle.SubText=A6L Events[0].Vehicle.SubBrand=5 Events[0].Vehicle.BrandYear=2 Events[0].TrafficCar.RelNDO=123 Events[0].TrafficCar.PlateNumber=AC00003 Events[0].TrafficCar.PlateColor=Yellow Events[0].TrafficCar.VehicleColor=Blue Events[0].TrafficCar.BoundingBox[0]=1341 Events[0].TrafficCar.BoundingBox[1]=2451 Events[0].TrafficCar.BoundingBox[2]=4513 Events[0].TrafficCar.BoundingBox[3]=4135 Events[0].TrafficCar.Country=China Events[0].CommlInfo.Seat[0].Type>Main

	<pre> Events[0].CommlInfo.Seat[0].Status[0]=Smoking Events[0].CommlInfo.Seat[0].SunShade=WithSunShade Events[0].CommlInfo.Seat[0].ShadePos[0]=2021 Events[0].CommlInfo.Seat[0].ShadePos[1]=3041 Events[0].CommlInfo.Seat[0].ShadePos[2]=2151 Events[0].CommlInfo.Seat[0].ShadePos[3]=3661 Events[0].CommlInfo.Seat[0].SafeBelt=WithoutSafeBelt ... --<boundary> Content-Type: image/jpeg Content-Length: <image size> <Jpeg image data> --<boundary></pre>
--	---

10.1.2 [Event] TrafficRetrograde

When detects vehicle retrograde, send this event.

Event params is the same as **TrafficJunction**, except for event Code is **TrafficRetrograde**.

10.1.3 [Event] TrafficJam

Table 10-2

Usage	Refer to "4.4.3 Subscribe to Snapshot"for how to subscribe event		
Description	When detects traffic jam, send this event.		
[Event Params] (key=value format)			
Name	Type	R/O	Param Description
Events	array<object>	R	Event info array.
+EventBaseInfo	object	R	Base info of event.
++Code	String	R	Event Code. It should be TrafficJam .
++Action	String	R	Event Action. It can be "Start", "Stop" or "Pulse".
++Index	int	O	The channel index relate to this event.
+GroupID	int	O	The id of event group.
+CountInGroup	int	O	Event count in the event group.
+IndexInGroup	int	O	The index of this event in the event group, start from 1.
+Lane	int	O	Lane number, start from 0.
+StartJaming	int	O	Start jam time, UTC seconds.
+AlarmInterval	int	O	Alarm interval, unit is second.
+JamLenght	int	O	Jam length, percentage of the lane.
+JamRealLength	int	O	Jam real length, unit is metre.
[Example]			
Event	<pre>--<boundary> Content-Type: text/plain Content-Length: <length></pre>		

```

Events[0].EventBaseInfo.Code=TrafficJam
Events[0].EventBaseInfo.Action=Pulse
Events[0].EventBaseInfo.Index=0
Events[0].GroupID=123
Events[0].CountInGroup=3
Events[0].IndexInGroup=1
Events[0].Lane=0
Events[0].StartJaming=123456789
Events[0].AlarmInterval=180
Events[0].JamLength=70
Events[0].JamRealLength=120
...
--<boundary>
Content-Type: image/jpeg
Content-Length: <image size>

<Jpeg image data>
--<boundary>

```

10.1.4 [Event] TrafficUnderSpeed

When detects vehicle under speed, send this event.

Event params is the same as **TrafficJunction**, except for event Code is **TrafficUnderSpeed**, and add following params:

Table 10-3

[Extra Event Params] (key=value format)			
Name	Type	R/O	Param Description
+SpeedLimit	array<int>	O	Speed limit, 2 integer, min speed and max speed.
+UnderSpeeding Percentage	int	O	Percentage of under speed.

10.1.5 [Event] TrafficOverSpeed

When detects vehicle over speed, send this event.

Event params is the same as **TrafficJunction**, except for event Code is **TrafficOverSpeed**, and add following params:

Table 10-4

[Extra Event Params] (key=value format)			
Name	Type	R/O	Param Description
+SpeedLimit	array<int>	O	Speed limit, 2 integer, min speed and max speed.
+SpeedingPercentage	int	O	Percentage of over speed.

10.1.6 [Event] TrafficPedestrain

Table 10-5

Usage	Refer to "4.4.3 Subscribe to Snapshot" for how to subscribe event		
Description	When detects human in the lane, send this event		
[Event Params] (key=value format)			
Name	Type	R/O	Param Description
Events	array<object>	R	Event info array.
+EventBaseInfo	object	R	Base info of event.
++Code	String	R	Event Code. It should be TrafficPedestrain .
++Action	String	R	Event Action. It can be "Start", "Stop" or "Pulse".
++Index	int	O	The channel index relate to this event.
+GroupID	int	O	The id of event group.
+CountInGroup	int	O	Event count in the event group.
+IndexInGroup	int	O	The index of this event in the event group, start from 1.
+Lane	int	O	Lane number, start from 0.
+Vehicle	object	O	Actually this is the human that detected, not vehicle.
++Category	string	R	Must be " Passerby ", means this actually is a human.
++BoundingBox	Array<int>	R	The detected car bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
[Example]			
Event	<pre>--<boundary> Content-Type: text/plain Content-Length: <length> Events[0].EventBaseInfo.Code=TrafficPedestrain Events[0].EventBaseInfo.Action=Pulse Events[0].EventBaseInfo.Index=0 Events[0].GroupID=123 Events[0].CountInGroup=3 Events[0].IndexInGroup=1 Events[0].Lane=0 Events[0].Vehicle.Category=Passerby Events[0].Vehicle.BoundingBox[0]=1341 Events[0].Vehicle.BoundingBox[1]=2451 Events[0].Vehicle.BoundingBox[2]=4513 Events[0].Vehicle.BoundingBox[3]=4135 ... --<boundary> Content-Type: image/jpeg Content-Length: <image size> <Jpeg image data></pre>		

--<boundary>

10.1.7 [Event] TrafficParking

When detects vehicle illegal parking, send this event.

Event params is the same as **TrafficJunction**, except for event Code is **TrafficParking**, and remove param "**speed**", add following params:

Table 10-6

[Extra Event Params] (key=value format)			
Name	Type	R/O	Param Description
+StartParking	int	O	The start time of parking, UTC seconds.
+AlarmInterval	int	O	The alarm interval.
+ParkingAllowedTime	int	O	The allowed time of parking.

10.2 Traffic Flow

10.2.1 [Event] TrafficFlowStat

Table 10-7

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When traffic flow trigger the rule, send this event		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
FlowStates	array<object>	R	Traffic flow info, each object in list is traffic flow info about one lane.
+Lane	int	R	Lane number, start from 0.
+Flow	int	R	Traffic flow number.
+Period	int	R	Traffic stat time, unit is minute.
+PeriodByMili	int	O	Traffic stat time, unit is millisecond, value should between 0 and 59999.
+DrivingDirection	array<string>	O	Driving direction, should be an array of three strings: <ul style="list-style-type: none">• 1st string: direction, can be: "Approach", "Leave"• 2nd string: Approach position name• 3rd string: Leave position name
[Example]			
Event	<p>--<boundary></p> <p>Content-Type: text/plain</p> <p>Content-Length: xxxx</p> <p>Code=TrafficFlowStat;action=Pulse;index=0;data={</p> <pre>"FlowStates": [{ "Lane": 0, "Flow": 50,</pre>		

	<pre> "Period": 5, "DrivingDirection": ["Approach", "XXXCity", "YYCity"] }, {...}, ... } --<boundary> </pre>
--	---

10.2.2 Find Traffic Flow History

Table 10-8

URL	http://<server>/cgi-bin/recordFinder.cgi?action=find&name=TrafficFlow		
Method	GET		
Description	Find traffic flow history record.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	The name of record table. It should be " TrafficFlow ".
count	int	O	Max result to return, default is 1024.
StartTime	string	O	The start of the record's CreateTime.
EndTime	string	O	The End of the record's CreateTime.
condition	object	O	Search condition.
+Channel	int	O	Video channel index which starts from 0.
+Lane	int	O	Lane index, starts from 0.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
totalCount	int	O	Total record num that find.
found	int	O	Record num that returned.
records	array<object>	R	The records that returned.
+RelNDo	int	R	Record id.
+CreateTime	int	R	The create time of record.
+StatisticsTime	int	R	Traffic flow statistics time, UTC seconds.
+Period	string	R	Statistics period, unit is second.
+Channel	int	R	Video channel index which starts from 0.
+Lane	int	R	Lane index, starts from 0.
+Vehicles	int	R	Total vehicle num.
+AverageSpeed	float	O	Average speed of the vehicle, -1 means no vehicle, 0 means congestion.
[Example]			
Request	GET http://192.168.1.250/cgi-bin/recordFinder.cgi?action=find&name=TrafficFlow&condition.Channel=0&condition.Lane=0&StartTime=123456700&EndTime=123456800&count=100		
Response	totalCount=1000 found=100 records[0].RelNDo=12345 records[0].CreateTime=123456789 records[0].StaticsticsTime=123456789 records[0].Period=300		

```

records[0].Channel=0
records[0].Lane=0
records[0].Vehicles=250
records[0].AverageSpeed=25.4
...
records[1].ReINDo=1357
9
records[1].CreateTime=123456799
records[1].StaticsticsTime=123456799
records[1].Period=300
records[1].Channel=0
records[1].Lane=0
records[1].Vehicles=220
records[1].AverageSpeed=21.8
...

```

10.2.3 Start Traffic Statistics Search

Request URL	http://<server>/cgi-bin/api/trafficFlowStat/startFind		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
condition	object	R	Traffic flow statistical conditions
+StartTime	systemtime	R	Start time, accurate to hour
+EndTime	systemtime	R	End time, accurate to hour
+Channel	int	O	Channel number
+Lane	int	O	Lane number
+Lanes	int[]	O	Lane number array, supports data search by multiple lanes
+ClassType	int	O	As video metadata and road monitoring use the same database table, they need to be distinguished according to their types (0 represents video metadata and 1 represents road monitoring).

+Granularity	enumchar[8]	O	<p>The granularity of statistical information that needs to be returned in the searching process. This needs to be supported by getCaps.</p> <pre>Enumchar[8]{ "Minute": By 5 minutes (hourly report) "Hour": By hour (daily report) "Day": By day (monthly report) "Month": By month (yearly report) }</pre> <p>The default value is Hour.</p>	"Hour"
+PresetID	int[]	O	Preset	[1]
+Direction	enumchar[16]	O	<p>Search by statistical direction</p> <pre>enumchar[16]{ "Both" "LeftToRight" "RightToLeft" }</pre>	"Both"
+Channels	int[256]	O	<p>Channel number array. Used in structured traffic statistics. In multi-channel query, the channel number is distributed to query the total data of all channels</p> <p>There must be at least one channel and channels field. Generally, they should not both exist. If both exist, channels shall prevail.</p>	[0,1,2,5]

Request Example

```
{
  "condition": {
    "StartTime": "2010-05-25 00:00:00",
    "EndTime": "2010-05-25 23:59:59",
    "Channel": 0,
    "Lane": 0,
    "Lanes": [0],
    "ClassType": 1,
    "Granularity": "Hour",
    "PresetID": [1],
    "Direction": "Both",
    "Channels": [0,1,2,5]
  }
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
------	------	-----	-------------	---------

token	uint	O	token	2342343
totalCount	uint	O	Total number of results that meet the query criteria	333

Response Example

```
{
    "token": 2342343,
    "totalCount": 333
}
```

10.2.4 Get Traffic Statistics

Request URL	http://<server>/cgi-bin/api/trafficFlowStat/doFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
token	uint	R	Search token.	46878
beginNumber	uint	R	Search begin number. That is to start from the beginNumber records, fetch the count records, and return. $0 \leqslant \text{beginNumber} \leqslant \text{totalCount}-1$	0
count	uint	R	Number of traffic statistic entries per search	24
Request Example				
{	<pre>"token": 46878, "beginNumber": 0, "count": 24</pre>			

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
found	int	R	Number of entries that are searched	12
info	Info[]	R	Traffic statistics (For details, see traffic-flow statistics status information structure in standard.)	[, ...]

Response Example

```
{
  "found": 12,
  "info": [
    {
      "PresetID" : 1
      "DrivingDirection" : ["Approach", "XXXCity", "YYCity"], "Lane"
      : 1,
      "MachineAddress" : "XXX District, YYY Road", "MachineName"
      : "Device001",
      "UTC" : 1465389203,
      "UTCMS" : 123,
      .....
    },...{}]
}
}
```

10.2.5 End Traffic Statistics Search

Request URL	http://<server>/cgi-bin/api/trafficFlowStat/stopFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
token	uint	R	Search token.	46878
Request Example				

```
{
  "token": 46878
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
------	------	-----	-------------	---------

Response Example

```
{}
```

10.3 Traffic Record

10.3.1 Insert Traffic BlockList/AllowList Record

Table 10-9

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=insert&name=<RecordName>		
Method	GET		
Description	Insert traffic BlockList/AllowList record.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	The record table name, " TrafficBlackList " for BlockList and " TrafficRedList " for AllowList.
PlateNumber	string	R	The number of car plate, max string length is 31. It must be unique.
MasterOfCar	string	O	The car owner, max string length is 15
PlateColor	string	O	Plate color, max string length is 31, ex: "Yellow", "Blue", ... etc.
PlateType	string	O	Plate type, max string length is 31.
VehicleType	string	O	Vehicle type, max string length is 31.
VehicleColor	string	O	Vehicle color, max string length is 31. ex: "Yellow", "Blue", ... etc.
BeginTime	string	O	Begin time, ex: "2010-05-25 00:00:00".
CancelTime	string	O	Cancel time, ex: "2010-06-25 00:00:00".
AuthorityList	object	O	Authority list, only valid for " TrafficRedList " table.
+OpenGate	bool	O	Authority to open the gate.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
RelINDo	int	R	The new record's id, return -1 if the device handles asynchronously.
[Example]			
Request	GET http://192.168.1.250/cgi-bin/recordUpdater.cgi?action=insert&name=TrafficBlackList&PlateNumber=AC00001&MasterOfCar=Rohit&PlateColor=Yellow&VehicleColor=Blue&BeginTime=2011-01-01%2012:00:00&CancelTime=2011-01-10%2012:00:00		
Response	RelINDo=12345		

10.3.2 Update Traffic BlockList/AllowList Record

Table 10-10

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=update&name=<RecordName>		
Method	GET		
Description	Update Traffic BlockList/AllowList record. Note: Besides action, name, reINDo, there should be at least one more parameter to be updated.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	The record table name, " TrafficBlackList " for BlockList and " TrafficRedList " for AllowList.
reINDo	int	R	record id
PlateNumber	string	R	The number of car plate, max string length is 31. It must be unique.
MasterOfCar	string	O	The car owner, max string length is 15
...<other param>	-	-	...<See above insert command for other params of the record. They are all optional.>
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.250/cgi-bin/recordUpdater.cgi?action=update&name=TrafficBlackList&reINDo=12345&PlateNumber=AC00001&MasterOfCar=Rohit&PlateColor=Yellow&VehicleColor=Blue&BeginTime=2011-01-01%2012:00:00&CancelTime=2011-01-10%2012:00:00		
Response	OK		

10.3.3 Remove Traffic BlockList/AllowList Record

Table 10-11

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=remove&name=<RecordName>		
Method	GET		
Description	Remove Traffic BlockList/AllowList record.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	The record table name, " TrafficBlackList " for BlockList and " TrafficRedList " for AllowList.
reINDo	int	R	The record id.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.250/cgi-bin/recordUpdater.cgi?action=remove&name=TrafficBlackList&reINDo=12345		

Response

OK

10.3.4 Find Traffic BlockList/AllowList Record

Table 10-12

URL	http://<server>/cgi-bin/recordFinder.cgi?action=find&name=<RecordName>		
Method	GET		
Description	Find Traffic BlockList/AllowList record.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	The record table name, "TrafficBlackList"for BlockList and "TrafficRedList"for AllowList.
count	int	O	Max result to return, default is 1024.
StartTime	string	O	The start of the record's CreateTime.
EndTime	string	O	The end of the record's CreateTime.
condition	object	O	Search condition.
+PlateNumber	string	O	Car plate number, max string length is 47.
+PlateNumberVague	string	O	Car plate number substring, match any car plate number that contain this substring, max string length is 47.
+PlateNumberVagueGroup	array<string>	O	Car plate number substring array, max string length is 47.
+QueryCount	int	O	Query count, default is 1000.
+QueryResultBegin	int	O	Begin number in the result set, default is 0.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
totalCount	int	O	Total record num that find.
found	int	O	Record num that returned.
records	array<object>	R	The records that returned.
+ReINDo	int	R	Record id.
+CreateTime	int	R	The create time of record.
+PlateNumber	string	R	Car plate number.
+MasterOfCar	string	O	Car owner.
+...<other param>	-	-	...<see above insert command for other params of the record>
[Example]			
Request	GET http://192.168.1.250/cgi-bin/recordFinder.cgi?action=find&name=TrafficBlackList&condition.PlateNumber=AC00001&StartTime=123456700&EndTime=123456800&count=100		
Response	totalCount=1000 found=100 records[0].ReINDo=12345 5 records[0].CreateTime=123456789 records[0].PlateNumber=AC00001		

	records[0].MasterOfCar=Rohit ... records[1].ReINDo=1357 9 records[1].CreateTime=123456799 records[1].PlateNumber=AC00001 records[1].MasterOfCar=LiSi ...
--	---

10.3.5 RemoveEx Traffic BlockList/AllowList Record

Table 10-13

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=removeEx&name=<RecordName>		
Method	GET		
Description	RemoveEx Traffic BlockList/AllowList record.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	The record table name, "TrafficBlackList" for BlockList and "TrafficRedList" for AllowList.
ReINDo	int	O	The record id.
PlateNumber	string	O	The number of car plate, max string length is 31. It must be unique.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.250/cgi-bin/recordUpdater.cgi?action=removeEx&name=TrafficBlackList &PlateNumber=AC00001		
Response	OK		

10.3.6 Import Traffic BlockList/AllowList

- Import the traffic blocklist and allowlist

Upload a blocklist or allowlist file to the device, and then import the data into the device. This operation might take a long time.

Request URL	http://<server>/cgi-bin/trafficRecord.cgi?action=uploadFile&Type=<Type>		
Method	POST		
Request Params (key=value format in URL ; binary data in body)			
Name	Type	R/O	Description
Type	char[16]	R	File type, value: "TrafficBlackList" Traffic blocklist "TrafficRedList" Traffic allowlist
format	char[16]	R	File format, fixed to "CSV"
code	char[16]	R	Encoding format, the value is "utf-8" or "GB2312"
Request Example			

```

POST http://192.168.1.250/cgi-bin/trafficRecord.cgi?action=uploadFile&Type=TrafficBlackList&format=CSV&code=utf-8 HTTP/1.1
User-Agent: Client/1.0
Content-Type: multipart/form-data; boundary=<boundary>
Content-Length:XXXX

--<boundary>
Content-Disposition: form-data; name="blackfile"; filename="TrafficBlackList.CSV"
Content-Type: application/vnd.ms-excel

<File data....>
--<boundary>--

```

Response Params (OK in body)

Response Example

OK

10.3.7 Export Traffic BlockList/AllowList

- Asynchronously export traffic blocklist and allowlist

Notify the device to export the traffic blocklist and allowlist to the file. This operation is non-blocking.
To obtain the export result, please run the "getFileExportState" command.

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=exportAsyncFile&name=<Name>			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	char[16]	R	File type, value: "TrafficBlackList" Traffic blocklist "TrafficRedList" Traffic allowlist	"TrafficBlackList"
filename	char[256]	O	Export Filename	"RecordFile01"
format	char[16]	R	File format, fixed to "CSV"	CSV
code	char[16]	R	Encoding format, the value is "utf-8" or "GB2312"	"utf-8"
Request Example				
http://<server>/cgi-bin/recordUpdater.cgi?action=exportAsyncFile&name=TrafficBlackList&filename=RecordFile01&format=CSV&code=utf-8				

Response Params (OK in body)

Response Example

OK

- Get the status of the exported traffic blocklist and allowlist

Get the result of the exported traffic blocklist and allowlist

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=getFileExportState&name=<Name>		
Method	GET		

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
name	char[16]	R	File type, value: "TrafficBlackList" Traffic blocklist "TrafficRedList" Traffic allowlist	"TrafficBlackList"

Request Example

http://<server>/cgi-bin/recordUpdater.cgi?action=getFileExportState&name=TrafficBlackList

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
state	int	R	Exporting status: 0: Success 1: Failure 2: running 3: the file is invalid 4: the file is too large 5: the file has duplicate data	1

Response Example

state=0

- Download the exported traffic blocklist and allowlist file

Download the exported traffic blocklist and allowlist file

Request URL http://<server>/cgi-bin/recordUpdater.cgi?action=downloadFile&Type=<Type>

Method GET

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
Type	char[16]	R	File type, value: "TrafficBlackList": Traffic blocklist "TrafficRedList": Traffic allowlist	"TrafficBlackList"
filename	char[256]	O	The exported file name must be the same as the file name used in the export command.	"RecordFile01"

Request Example

http://<server>/cgi-bin/recordUpdater.cgi?action=downloadFile&Type=TrafficBlackList&filename=RecordFile01

Response Params (binary in body)

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: application/octet-stream

Content-Length: xxx

<File Data...>

10.3.8 Export Traffic Flow

- Export traffic flow records

Notify the device to export the traffic flow records to the file. This operation is non-blocking. To obtain the export result, please run the "getFileExportState" command.

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=exportAsyncFile&name=TrafficFlow			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	char[16]	R	File type, fixed value: "TrafficFlow"	"TrafficFlow"
filename	char[256]	O	Name of the exported file	"RecordFile01"
format	char[16]	R	Format of the exported file, fixed to "CSV"	CSV
code	char[16]	R	Encoding format of the exported file, the value is "utf-8" or "GB2312"	"utf-8"

Request Example

```
http://<server>/cgi-bin/recordUpdater.cgi?action=exportAsyncFile&name=TrafficFlow&filename=RecordFile01&format=CSV&code=utf-8
```

Response Params (OK in body)

Response Example

```
OK
```

- Get the exported traffic flow record results

Get the exported traffic flow record results

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=getFileExportState&name=TrafficFlow			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	char[16]	R	File type, fixed value: "TrafficFlow"	"TrafficFlow"

Request Example

```
http://<server>/cgi-bin/recordUpdater.cgi?action=getFileExportState&name=TrafficFlow
```

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
state	int	R	Export status, the value is 0: success, 1: failure, 2: running, 3: the file is invalid, 4: the file is too large, 5: the file has duplicate data	1

Response Example

```
state=0
```

- Download Export 0 traffic flow records File

Download the exported traffic flow log file

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=downloadFile&Type=TrafficFlow
--------------------	--

Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Type	char[16]	R	File type, fixed value: "TrafficFlow"	"TrafficFlow"
filename	char[256]	O	The exported file name must be the same as the file name used in the export command.	"RecordFile01"
Request Example				
http://<server>/cgi-bin/recordUpdater.cgi?action=downloadFile&Type=TrafficFlow&filename=RecordFile01				

Response Params (binary in body)
Response Example
HTTP/1.1 200 OK
Server: Device/1.0
Content-Type: application/octet-stream
Content-Length: xxx
<File Data...>

10.3.9 Export Traffic Snap Event Info

- Export traffic snapshot event records

Notify the device to export the traffic snapshot event records to the file. This operation is non-blocking. To obtain the export result, please run the “getFileExportState” command.

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=exportAsyncFileByConditon&name=TrafficSnapEventInfo			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	char[16]	R	File type, fixed value: "TrafficSnapEventInfo"	"TrafficSnapEventInfo"
filename	char[256]	O	Export Filename	"RecordFile01"
format	char[16]	R	File format, fixed to "CSV"	CSV
code	char[16]	R	Encoding format, the value is "utf-8" or "GB2312"	"utf-8"
condition	object	R	Data Conditions	"utf-8"
+startTime	char[16]	R	Export the Start time, format: yyyy-MM-dd HH:mm:ss	"2010-04-08 16:12:46"
+endTime	char[16]	R	Export the End time, format: yyyy-MM-dd HH:mm:ss	"2010-04-08 18:12:46"
Request Example				
http://<server>/cgi-bin/recordUpdater.cgi?action=exportAsyncFileByConditon&name=TrafficSnapEventInfo&filename=RecordFile01&format=CSV&code=utf-8&condition.startTime=2010-04-08%2016:12:46&condition.endTime=2010-04-08%2018:12:46				

Response Params (OK in body)**Response Example**

OK

- Obtain and export traffic snapshot event record results

Obtain and export traffic snapshot event record results

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=getFileExportState&name=TrafficSnapshotEventInfo		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
name	char[16]	R	File type, fixed value: "TrafficSnapEventInfo"
Request Example			
http://<server>/cgi-bin/recordUpdater.cgi?action=getFileExportState&name=TrafficSnapEventInfo			

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
state	int	R	Export status, the value is 0: success, 1: failure, 2: running, 3: the file is invalid, 4: the file is too large, 5: the file has duplicate data	1
Response Example				
state=0				

- Download the exported traffic snapshot event record file

Download the exported traffic snapshot event record file

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=downloadFile&Type=TrafficSnapshotEventInfo		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Type	char[16]	R	File type, fixed value: "TrafficSnapEventInfo"
filename	char[256]	O	The exported file name must be the same as the file name used in the export command.
Request Example			
http://<server>/cgi-bin/recordUpdater.cgi?action=downloadFile&Type=TrafficSnapshotEventInfo&filename=RecordFile01			

Response Params (binary in body)**Response Example**

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: application/octet-stream

Content-Length: xxx

<File Data...>

10.4 Traffic Snap Operation

10.4.1 Open Strobe

Table 10-14

Syntax	http://<server>/cgi-bin/trafficSnap.cgi?action=openStrobe&channel=< ChannelNo >&info.openType=< OpenType >&info.plateNumber=< PlateNumber >
Method	GET
Description	Open the strobe manually.
Example	http://192.168.1.250/cgi-bin/trafficSnap.cgi?action=openStrobe&channel=1&info.openType=Normal&info.plateNumber=046XRW
Success Return	OK
Comment	Parameters in URL: ChannelNo : integer, video channel index which starts from 1. OpenType : string, For now, the value is fixed to "Normal" PlateNumber : string, the plateNumber of the TrafficCar.

10.4.2 Open/Close Unlicensed Vehicle Detection

Table 10-15

Syntax	http://<server>/cgi-bin/trafficSnap.cgi?action=< Action >&channel=< ChannelNo >&name=UnlicensedVehicle
Method	GET
Description	Open or Close the unlicensed vehicle detection.
Example	http://192.168.1.250/cgi-bin/trafficSnap.cgi?action=open&channel=1&name=UnlicensedVehicle
Success Return	OK
Comment	Parameters in URL: Action : string, this value can be open and close. ChannelNo : integer, video channel index which starts from 1.

10.4.3 Manual Snap

Table 10-16

Syntax	http://<server>/cgi-bin/trafficSnap.cgi?action=manSnap&channel=< ChannelNo >
Method	GET
Description	Take a snapshot manually. For intelligent traffic device, it should use this method to take a snapshot.

	<p>But, the response is not image data.If you want to get the image data, please follow these steps:</p> <ol style="list-style-type: none"> 1. Use the method mentioned chapter (4.4.3 Subscribe to snapshot) to subscribe the image data, and the eventcode is "TrafficManualSnap". 2. Use the "manSnap"to take a snapshot manually. 3. In the connection which built in the Step 1, the device will send the image data.
Example	http://192.168.1.250/cgi-bin/trafficSnap.cgi?action=manSnap&channel=1
Success Return	OK
Comment	<p>Parameters in URL: ChannelNo: integer, video channel index which starts from 1.</p>

10.5 Traffic Parking

10.5.1 Get the Specific Parking Space Status

Table 10-17

Syntax	<a href="http://<server>/cgi-bin/trafficSnap.cgi?action=getParkingSpaceStatus&channel=<ChannelNo>&<paramName>=<paramValue>[&<paramName>=<paramValue>...>]">http://<server>/cgi-bin/trafficSnap.cgi?action=getParkingSpaceStatus&channel=<ChannelNo>&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Get specific parking space(s) status.
Example	http://192.168.1.250/cgi-bin/trafficSnap.cgi?action=getParkingSpaceStatus&condition.Lane[0]=0&condition.Lane[1]=255
Success Return	<p>A list of parking space status status[0].Lane=0 status[0].PictureId=5 status[0].TrafficCar.CountInGroup=1 ... status[1].Lane=1 status[1].PictureId=4 status[1].TrafficCar.CountInGroup=1 ...</p>
Comment	<p>Parameters in URL: ChannelNo: integer, video channel index which starts from 1. paramName and paramValue: detail in table below.</p> <p>In table below, index: The index of type array, start from 0</p> <p>Parameters in Response : TrafficCar: the members refer to TrafficCar</p>

Appendix

ParamName	ParamValue type	Description
-----------	-----------------	-------------

ParamName	ParamValue type	Description
condition.Lane[<i>index</i>]	int	The Lane value
condition.ResponseLevel	int	The Level value , refer to condition

10.5.2 Get All Status of Parking Spaces

Table 10-18

Syntax	http://<server>/cgi-bin/trafficParking.cgi?action=getAllParkingSpaceStatus
Method	GET
Description	Get all valid parking spaces status of one device.
Example	http://192.168.1.250/cgi-bin/trafficParking.cgi?action=getAllParkingSpaceStatus
Success Return	<pre> status[0].Lane=0 status[0].CustomParkNo = A2701 status[0].Status = Park status[0].SpaceType = 0 ... status[1].Lane=1 status[1].Status = NoPark ... sceneType = ParkingStatistics statisticsMode = AreaMode areaStatus[0].Name = A01 areaStatus[0].Count = 10 areaStatus[0].SpaceType = 0 areaStatus[0].RemainINDt = 5 ... areaStatus[1].Name = A02 areaStatus[1].Count = 20 ... </pre>
Comment	<p>Parameters in Response :</p> <p>Status : Park or NoPark</p> <p>CustomParkNo: customized parking space number</p> <p>SpaceType : 0: trolley parking space, 1: Crane parking space</p> <p>sceneType : scene type; "parkingspace": parking space detection type; and "parkingstatistics": parking space statistics type</p> <p>statisticsMode : Statistical mode, valid when the sceneType is "parkingstatistics". "areamode": area mode, "spacemode": parking space mode</p> <p>areaStatus : area status. It is an array. It is valid when scenetype is "parkingstatistics" and statisticsmode is "areamode"</p> <p>SpaceType : parking space type: 0: trolley parking space, 1: Crane parking space</p>

10.5.3 [Config] Parking Space Light State

- Get parking space light state

Table 10-19

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=ParkingSpaceLightState
Method	GET
Description	Use this method. It can get the light state config. For example, it can know that when the space is free, then the light should be green, and the space is full, the light should be red.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=ParkingSpaceLightState
Success Return	<pre>table.ParkingSpaceLightState.SpaceFree.Blue=0 table.ParkingSpaceLightState.SpaceFree.Green=1 table.ParkingSpaceLightState.SpaceFree.Pink=0 table.ParkingSpaceLightState.SpaceFree.Purple=0 table.ParkingSpaceLightState.SpaceFree.Red=0 table.ParkingSpaceLightState.SpaceFree.White=0 table.ParkingSpaceLightState.SpaceFree.Yellow=0 table.ParkingSpaceLightState.SpaceFull.Blue=0 table.ParkingSpaceLightState.SpaceFull.Green=0 table.ParkingSpaceLightState.SpaceFull.Pink=0 table.ParkingSpaceLightState.SpaceFull.Purple=0 table.ParkingSpaceLightState.SpaceFull.Red=1 table.ParkingSpaceLightState.SpaceFull.White=0 table.ParkingSpaceLightState.SpaceFull.Yellow=0 table.ParkingSpaceLightState.SpaceOrder.Blue=0 table.ParkingSpaceLightState.SpaceOrder.Green=0 table.ParkingSpaceLightState.SpaceOrder.Pink=0 table.ParkingSpaceLightState.SpaceOrder.Purple=0 table.ParkingSpaceLightState.SpaceOrder.Red=0 table.ParkingSpaceLightState.SpaceOrder.White=0 table.ParkingSpaceLightState.SpaceOrder.Yellow=1 table.ParkingSpaceLightState.SpaceOverLine.Blue=0 table.ParkingSpaceLightState.SpaceOverLine.Green=0 table.ParkingSpaceLightState.SpaceOverLine.Pink=0 table.ParkingSpaceLightState.SpaceOverLine.Purple=0 table.ParkingSpaceLightState.SpaceOverLine.Red=0 table.ParkingSpaceLightState.SpaceOverLine.White=0 table.ParkingSpaceLightState.SpaceOverLine.Yellow=1 table.ParkingSpaceLightState.SpaceSpecial.Blue=0 table.ParkingSpaceLightState.SpaceSpecial.Green=0 table.ParkingSpaceLightState.SpaceSpecial.Pink=0 table.ParkingSpaceLightState.SpaceSpecial.Purple=0 table.ParkingSpaceLightState.SpaceSpecial.Red=0 table.ParkingSpaceLightState.SpaceSpecial.White=0 table.ParkingSpaceLightState.SpaceSpecial.Yellow=1</pre>
Comment	<p>Parameters in Response :</p> <p>There are five conditions about the space light state: SpaceFree, SpaceFull, SpaceOrder, SpaceOverLine, and SpaceSpecial. For each condition, you can choose the light state.</p>

	The state value is integer. It can be 0, 1, 2. 0 means close; 1 means open; 2 means twinkle.
--	---

- Set parking space light state

Table 10-20

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set the light state config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&ParkingSpaceLightState.SpaceFree.Blue=1&ParkingSpaceLightState.SpaceFree.Green=0&ParkingSpaceLightState.SpaceFree.Pink=0&ParkingSpaceLightState.SpaceFree.Purple=0&ParkingSpaceLightState.SpaceFree.Red=0&ParkingSpaceLightState.SpaceFree.White=0&ParkingSpaceLightState.SpaceFree.Yellow=0
Success Return	OK
Comment	—

10.5.4 Set Order State

Table 10-21

Syntax	http://<server>/cgi-bin/trafficParking.cgi?action=setOrderState&state[0].Lane=<LaneNumber>&state[0].State=<State>
Method	GET
Description	Set the light order state, order or not
Example	http://192.168.1.250/cgi-bin/trafficParking.cgi?action=setOrderState&state[0].Lane=0&state[0].State=Ordered
Success Return	OK
Comment	Parameters in URL: LaneNumber : integer, for now, fixed to 0. State : string, the value can be "Ordered"or "Free".

10.5.5 Set Light State

Table 10-22

Syntax	http://<server>/cgi-bin/trafficParking.cgi?action=setLightState&state[Index].LightNo=<LaneNumber>&state[Index].Color=<Color>&state[Index].State=<State>&state[Index].Enable=<Enable>
Method	GET
Description	Set the light state.
Example	http://192.168.1.250/cgi-bin/trafficParking.cgi?action=setLightState&state[0].LightNo=0

	&state[0].Color=Red&state[0].State=0&state[0].Enable=true
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>Index: integer, start with 0.</p> <p>LaneNumber: integer, the No. of the Lane.</p> <p>Color: string, it can be Red, Yellow, Blue, Green, Purple, White, Pink.</p> <p>State: integer, it can be 0, 1, 2.0 means close;1 means open;2 means twinkle;</p> <p>Enable: true or false, enable or not.</p>

10.5.6 [Config] Parking Space Access Filter Setting

- Get parking space access filter setting

Table 10-23

URL	http://<server>/cgi-bin/ConfigManager.cgi?action=getConfig&name=ParkingSpaceAccessFilter					
Method	GET					
Description	Get Parking Space Access Filter config. Using this method, we can get the accessible address of the device.					
[Request Params] (None)						
[Response Params] (key=value format)						
Name	Type	R/O	Param Description			
table	object	R	config table object			
+ParkingSpaceAccessFilter	object	R	ParkingSpaceAccessFilter config object			
++Enable	bool	R	Enable filter or not			
++Type	string	R	Filter type, can be: "BannedList", "TrustList".			
++TrustList	Array<string>	O	IP address list that trust			
++BannedList	Array<string>	O	IP address list that banned.			
[Example]						
Request	GET http://10.0.0.8/cgi-bin/ConfigManager.cgi?action=getConfig&name=ParkingSpaceAccessFilter					
Response	table.ParkingSpaceAccessFilter.Enable=false table.ParkingSpaceAccessFilter.Type=TrustList table.ParkingSpaceAccessFilter.TrustList[0]=172.24.2.14 table.ParkingSpaceAccessFilter.BannedList[0]=172.24.2.15					

- Set parking space access filter setting

Table 10-24

URL	http://<server>/cgi-bin/ConfigManager.cgi?action=setConfig
Method	GET
Description	Set Parking Space Access Filter config.

[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
ParkingSpaceAccessFilter	object	R	ParkingSpaceAccessFilter config object
+Enable	bool	R	Enable filter or not
+Type	string	R	Filter type, can be: "BannedList", "TrustList".
+TrustList	Array<string>	O	IP address list that trust
+BannedList	Array<string>	O	IP address list that banned.
[Response Params] (OK)			
[Example]			
Request	GET http://10.0.0.8/cgi-bin/ConfigManager.cgi?action=setConfig&ParkingSpaceAccessFilter.Enable=true&ParkingSpaceAccessFilter.Type=TrustList&ParkingSpaceAccessFilter.TrustList[0]=172.24.2.14&ParkingSpaceAccessFilter.BannedList[0]=172.24.2.15		
Response	OK		

10.5.7 Set OverLine State

Table 10-25

Syntax	http://<server>/cgi-bin/trafficParking.cgi?action=setOverLineState&state[0].Lane=<LaneNumber>&state[0].State=<State>
Method	GET
Description	Set the spaceState overLine state or StopOverLine state
Example	http://192.168.1.250/cgi-bin/trafficParking.cgi?action=setOverLineState&state[0].Lane=0&state[0].State=OverLine
Success Return	OK
Comment	Parameters in URL: LaneNumber : integer, for now, fixed to 0. State : string, the value can be "OverLine" or "StopOverLine".

10.6 Vehicles Distribution

10.6.1 Subscribe Vehicles Distribution Data

Table 10-26

URL	http://<server>/cgi-bin/vehiclesDistribution.cgi?action=attach		
Method	GET		
Description	Subscribe the vehicles Distribution data.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
Channel	int	R	Video channel index which starts from 1

heartbeat	int	O	Send heartbeat interval, range is [1, 60], unit is second. If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client, the heartbeat message are "Heartbeat". If this parameter is not present, its default value is 60.
-----------	-----	---	---

[Response Params] (key=value format)

Name	Type	R/O	Param Description
Channel	int	R	Video channel index which starts from 1
VehiclesData	array<object>	R	Vehicles distribution data info.
+PtzPresetId	int	R	The ptz preset index which starts from 1.
+RuleId	int	R	The rule id.
+RuleType	int	R	The rule type, 197 : vehicle congestion detection, 198 : vehicle limit detection.
+VehiclesNum	int	R	The vehicles number.
+QueueLen	int	R	The vehicle queue length.
+Region	array<array<int>>	R	The detection region, the first array is point list, max item is 32, the second array is point, must be two int, means x and y value, coordinate remap to 0 - 8192.

[Example]

Request	GET http://192.168.1.250/cgi-bin/vehiclesDistribution.cgi?action=attach&Channel=1&heartbeat=5
Response	<p>HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<boundary> Connection: closed</p> <p>--<boundary> Content-Type: text/plain Content-Length: <length></p> <p>Channel=1 VehiclesData[0].PtzPresetId=1 VehiclesData[0].RuleId=1 VehiclesData[0].RuleType=197 VehiclesData[0].VehiclesNum=25 VehiclesData[0].QueueLen=53 VehiclesData[0].Region[0][0]=1032 VehiclesData[0].Region[0][1]=1035 VehiclesData[0].Region[1][0]=1045 VehiclesData[0].Region[1][1]=5072 VehiclesData[0].Region[2][0]=6163 VehiclesData[0].Region[2][1]=5127 VehiclesData[0].Region[3][0]=6031 VehiclesData[0].Region[3][1]=1063</p> <p>--<boundary></p>

	Content-Type: text/plain Content-Length: 11
	Heartbeat -<boundary> Content-Type: text/plain Content-Length: <length>
	Channel=1 VehiclesData[0].PtzPresetId=2 ...

10.7 Vehicle Manager

10.7.1 Adding Vehicle Groups

Add vehicle groups, and then add vehicles to the groups.

Request URL	http://<server>/cgi-bin/api/VehicleRegisterDB/createGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
group	object	R	Group information	
+GroupName	char[128]	R	Group name	"Serious criminal database"
+GroupDetail	char[256]	O	Remarks of the group	"Key criminals"
+GroupType	enumchar[16]	O	Group type enumchar[16]{ "AllowListDB": Allowlist "BlockListDB": Blocklist }	"AllowListDB"
Request Example				
{ "group": { "GroupName": "Serious criminal database", "GroupDetail": "Key criminals", "GroupType": " AllowListDB " } }				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
groupId	char[64]	R	Group ID assigned by server	"10000"

Response Example

```
{  
    "groupID": "10000"  
}
```

10.7.2 Modifying Vehicle Groups

Request URL	http://<server>/cgi-bin/api/VehicleRegisterDB/modifyGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
group	object	R	Group information	
+GroupID	char[128]	R	Group ID	"0001"
+GroupName	char[128]	O	Group name	"Serious criminal database"
+GroupDetail	char[256]	O	Remarks of the group	"Key criminals"
+GroupType	enumchar[16]	O	Group type enumchar[16]{ "AllowListDB": Allowlist "BlockListDB": Blocklist }	" AllowListDB "
Request Example				
{ "group": { "GroupID": "0001", "GroupName": "Serious criminal database", "GroupDetail": "Key criminals", "GroupType": " AllowListDB " } }				

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

10.7.3 Deleting Vehicle Groups

Request URL	http://<server>/cgi-bin/api/VehicleRegisterDB/deleteGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
groupId	char[64]	R	Group ID. "" means to delete all groups.	"000001"
Request Example				
{ "groupId": "000001"				

```
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

10.7.4 Searching for Vehicle Groups

Request URL	http://<server>/cgi-bin/api/VehicleRegisterDB/findGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
groupId	char[64]	R	Group ID. “” means to search for all group information.	“000001”
Request Example				
{				
"groupId": "000001"				
}				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
GroupList	object[]	R	List of searched vehicle group information	
+groupId	char[64]	R	Group ID	“000001”
+groupName	char[128]	O	Group name	“Serious criminal database”
+groupDetail	char[256]	O	Remarks of the group	“Vehicles of key criminals”
+groupType	enumchar[16]	O	Group type enumchar[16]{ "AllowListDB": Allowlist "BlockListDB": Blocklist }	“AllowListDB”
+groupSize	int	O	Number of vehicles in the current group	30
+channels	int16[1024]	O	List of video channel numbers to which the current group is bound. If the current group is not bound to any video channel, enter [-1] in this parameter.	[0,]

Response Example				
{				
"GroupList": [
"groupID": "000001",				
"groupName": "Serious criminal database",				

```

        "groupDetail": "Vehicles of key criminals",
        "groupType": "AllowListDB",
        "groupSize": 30,
        "channels": [0]
    },...{}]
}

```

10.7.5 Adding Vehicle Records

Add vehicle records to vehicle groups.

Request URL	http://<server>/cgi-bin/api/VehicleRegisterDB/multiAppend		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
vehicle	object[]	R	Multiple vehicle information
+UID	uint64	R	Unique identifier of the vehicle, generated by the server to represent a unique identifier (primary key) in the program. It is different from the ID field.
+GroupID	char[64]	R	Group ID to which the vehicle belongs
+GroupName	char[128]	O	Group name to which the vehicle belongs (exists only when the parameter exists)
+PlateNumber	char[64]	O	License plate number
+PlateCountry	char[3]	O	The country where the vehicle is located (2 bytes), conforming to ISO3166.
+PlateType	uint16	O	License plate type
+Type	uint16	O	Vehicle type (such as sedans and trucks)
+Brand	uint16	O	Vehicle logo. You need to get the real logo through the mapping table, which is the same as CarLogoIndex of ANPR events.
+Serie	uint16	O	Vehicle sub-brand. You need to get the real sub-brand through the mapping table, which is the same as the SubBrand of ANPR event.
+CarSeriesMode YearIndex	uint16	O	Vehicle model year. You need to get the real model year through the mapping table, which is the same as

			BrandYear of ANPR event.. The model year No. of vehicle head ranges from 1 to 999; the model year No. of vehicle tail ranges 1001 to 1999. 0 means unknown, and 1000 is reserved.	
+VehicleColor	uint8[4]	O	Vehicle color. The first element represents the red component value. The second element represents the green component value. The third element represents the blue component value. The fourth element represents the transparency component value (meaningless).	[128, 128, 128, 255]
+VehicleColorState	uint8	O	Vehicle color status. 0: Unknown; 1: Known.	0
+PlateColor	uint8[4]	O	License plate color. It is the same as vehicle color.	[128, 128, 128, 255]
+PlateColorState	uint8	O	License plate color status. 0: Unknown; 1: Known.	0
+Name	char[64]	O	Name of the vehicle owner	"Rohit"
+Sex	char[16]	O	Gender "Male" "Female" "Unknown" (such as no field)	"Male"
+CertificateType	char[16]	O	ID card type IC: IC card Passport Officer: Officer card Unknown	"IC"
+ID	char[32]	O	Person ID, employee ID, and other IDs	"123456789"
+Country	char[3]	O	Nationality of vehicle owner (2 bytes), conforming to ISO3166.	"IND"
+Province	char[64]	O	Province	"XXX"
+City	char[64]	O	City	"YYY"
+HomeAddress	char[128]	O	Home address of the registered person (IVSS requirement)	"ZZZ Road"
+Email	char[32]	O	Email address of the vehicle owner	"user@example.com"

+PhoneNo	char[128]	O	Phone No. of the registered vehicle owner	"13xxxxx5678"
replace	bool	O	Overwrite the same record True: Yes; false: No.	true

Request Example

```
{
  "vehicle": [
    {
      "UID" : 120837,
      "GroupID" : "00001",
      "GroupName" : "Group1",
      "PlateNumber" : "ZA12345",
      "PlateCountry" : "IND",
      "PlateType" : 12,
      "Type" : 2,
      "Brand" : 10,
      "Serie" : 1005,
      "CarSeriesModelYearIndex" : 12,
      "VehicleColor" : [128, 128, 128, 255],
      "VehicleColorState" : 0,
      "PlateColor" : [128, 128, 128, 255],
      "PlateColorState" : 0,
      "Name" : "Rohit",
      "Sex" : "Male",
      "CertificateType" : "IC",
      "ID" : "123456789",
      "Country" : "IND",
      "Province" : "XXX",
      "City" : "YYY",
      "HomeAddress" : "ZZZ Road",
      "Email" : "user@example.com",
      "PhoneNo" : "13xxxxx5678",
      },...{}
    "replace": true
  }
}
```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
ErrCode	uint[]	O	Error code. Defined the hexadecimal error code before you use the error code. (Available for new devices). 0x0: Operation successful. 0x11340200: Database operation failed. 0x11340202: The license plate do not exist.	[0, 0]

		0x11340205: Exceeded the maximum number of license plate database.	
Response Example			
{		"ErrCode": [0, 0]	}

10.7.6 Modifying Vehicle Information

Request URL	http://<server>/cgi-bin/api/VehicleRegisterDB/modifyVehicle		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
vehicle	object	R	Vehicle information
+UID	uint64	R	Unique identifier of the vehicle, generated by the server to represent a unique identifier (primary key) in the program. It is different from the ID field.
+GroupID	char[64]	R	Group ID to which the vehicle belongs
+GroupName	char[128]	O	Group name to which the vehicle belongs (exists only when the parameter exists)
+PlateNumber	char[64]	O	License plate number
+PlateCountry	char[3]	O	The country where the vehicle is located (2 bytes), conforming to ISO3166.
+PlateType	uint16	O	License plate type
+Type	uint16	O	Vehicle type (such as sedans and trucks)
+Brand	uint16	O	Vehicle logo. You need to get the real logo through the mapping table, which is the same as CarLogoIndex of ANPR events.
+Serie	uint16	O	Vehicle sub-brand. You need to get the real sub-brand through the mapping table, which is the same as the SubBrand of ANPR

			event.	
+CarSeriesModelYearIndex	uint16	O	<p>Vehicle model year. You need to get the real model year through the mapping table, which is the same as BrandYear of ANPR event..</p> <p>The model year No. of vehicle head ranges from 1 to 999; the model year No. of vehicle tail ranges 1001 to 1999. 0 means unknown, and 1000 is reserved.</p>	12
+VehicleColor	uint8[4]	O	<p>Vehicle color.</p> <p>The first element represents the red component value.</p> <p>The second element represents the green component value.</p> <p>The third element represents the blue component value.</p> <p>The fourth element represents the transparency component value (meaningless).</p>	[128, 128, 128, 255]
+VehicleColorState	uint8	O	Vehicle color status. 0: Unknown; 1: Known.	0
+PlateColor	uint8[4]	O	License plate color. It is the same as vehicle color.	[128, 128, 128, 255]
+PlateColorState	uint8	O	License plate color status. 0: Unknown; 1: Known.	0
+Name	char[64]	O	Name of the vehicle owner	"Rohit"
+Sex	char[16]	O	<p>Gender</p> <p>"Male"</p> <p>"Female"</p> <p>"Unknown" (such as no field)</p>	"Male"
+CertificateType	char[16]	O	<p>ID card type</p> <p>IC: IC card</p> <p>Passport</p> <p>Officer: Officer card</p>	"IC"

			Unknown	
+ID	char[32]	O	Person ID, employee ID, and other IDs.	"13xxxxx6789"
+Country	char[3]	O	Nationality of vehicle owner (2 bytes), conforming to ISO3166.	"IND"
+Province	char[64]	O	Province	"XXX"
+City	char[64]	O	City	"YYY"
+HomeAddress	char[128]	O	Home address of the registered person (IVSS requirement)	"ZZZ Road"
+Email	char[32]	O	Email address of the vehicle owner	"user@example.com"
+PhoneNo	char[128]	O	Phone No. of the registered vehicle owner.	"13xxxxx5678"

Request Example

```
{
  "vehicle": {
    "UID" : 120837,
    "GroupID" : "00001",
    "GroupName" : "Group1",
    "PlateNumber" : "ZA12345",
    "PlateCountry" : "IND",
    "PlateType" : 12,
    "Type" : 2,
    "Brand" : 10,
    "Serie" : 1005,
    "CarSeriesModelYearIndex" : 12,
    "VehicleColor" : [128, 128, 128, 255],
    "VehicleColorState" : 0,
    "PlateColor" : [128, 128, 128, 255],
    "PlateColorState" : 0,
    "Name" : "Rohit",
    "Sex" : "Male",
    "CertificateType" : "IC",
    "ID" : "123456789",
    "Country" : "IND",
    "Province" : "XXX",
    "City" : "YYY",
    "HomeAddress" : "ZZZ Road",
    "Email" : "user@example.com",
    "PhoneNo" : "13xxxxx5678",
  }
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

10.7.7 Deleting Vehicle Information

Request URL	http://<server>/cgi-bin/api/VehicleRegisterDB/deleteVehicle			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
vehicle	object	R		
+UID	uint64	R	Unique vehicle identifier. It can also be deleted by groupID and plateNumber if UID is not provided.	124
+groupID	char[64]	R	Vehicle group ID	"3"
+plateNumber	char[64]	O	Licence plate	"ZA12345"
Request Example				
<pre>{ "vehicle": { "UID": 124, "groupID": "3", "plateNumber": "ZA12345" } }</pre>				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				
{}				

10.7.8 Searching for Vehicles from Registered Database

Request URL	http://<server>/cgi-bin/api/VehicleRegisterDB/startFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
vehicle	object	R	Vehicle information. GroupID is required. Other vehicle descriptions are optional, and you can search for vehicles with these descriptions from specific databases.	
+UID	uint64	R	Unique identifier of the vehicle, generated by the server to represent a unique identifier (primary key) in the program. It is	120837

			different from the ID field.	
+GroupId	char[64]	R	Group ID to which the vehicle belongs	"00001"
+GroupName	char[128]	O	Group name to which the vehicle belongs (exists only when the parameter exists)	"Group1"
+PlateNumber	char[64]	O	License plate number	"ZA12345"
+PlateCountry	char[3]	O	The country where the vehicle is located (2 bytes), conforming to ISO3166.	"IND"
+PlateType	uint16	O	License plate type	12
+Type	uint16	O	Vehicle type (such as sedans and trucks)	2
+Brand	uint16	O	Vehicle logo. You need to get the real logo through the mapping table, which is the same as CarLogoIndex of ANPR events.	10
+Serie	uint16	O	Vehicle sub-brand. You need to get the real sub-brand through the mapping table, which is the same as the SubBrand of ANPR event.	1005
+CarSeriesModelYearIndex	uint16	O	Vehicle model year. You need to get the real model year through the mapping table, which is the same as BrandYear of ANPR event.. The model year No. of vehicle head ranges from 1 to 999; the model year No. of vehicle tail ranges 1001 to 1999. 0 means unknown, and 1000 is reserved.	12
+VehicleColor	uint8[4]	O	Vehicle color. The first element represents the red component value. The second element represents the green component value. The third element represents the blue component value. The fourth element represents the transparency component value (meaningless).	[128, 128, 128, 255]
+VehicleColorState	uint8	O	Vehicle color status. 0: Unknown; 1: Known.	0
+PlateColor	uint8[4]	O	License plate color. It is the same as vehicle color.	[128, 128, 128, 255]
+PlateColorState	uint8	O	License plate color status. 0: Unknown; 1: Known.	0

+Name	char[64]	O	Name of the vehicle owner	"Rohit"
+Sex	char[16]	O	Gender "Male" "Female" "Unknown" (such as no field)	"Male"
+CertificateType	char[16]	O	ID card type IC: IC card Passport Officer: Officer card Unknown	"IC"
+ID	char[32]	O	Person ID, employee ID, and other IDs.	"123456789"
+Country	char[3]	O	Nationality of vehicle owner (2 bytes), conforming to ISO3166.	"IND"
+Province	char[64]	O	Province	"XXX"
+City	char[64]	O	City	"YYY"
+HomeAddress	char[128]	O	Home address of the registered person (IVSS requirement)	"ZZZ Road"
+Email	char[32]	O	Email address of the vehicle owner	"user@example.com"
+PhoneNo	char[128]	O	Phone No. of the registered vehicle owner.	"13xxxxx5678"

Request Example

```
{
  "vehicle": {
    "UID" : 120837,
    "GroupID" : "00001",
    "GroupName" : "Group1",
    "PlateNumber" : "ZA12345",
    "PlateCountry" : "IND",
    "PlateType" : 12,
    "Type" : 2,
    "Brand" : 10,
    "Serie" : 1005,
    "CarSeriesModelYearIndex" : 12,
    "VehicleColor" : [128, 128, 128, 255],
    "VehicleColorState" : 0,
    "PlateColor" : [128, 128, 128, 255],
    "PlateColorState" : 0,
    "Name" : "Rohit",
    "Sex" : "Male",
    "CertificateType" : "IC",
    "ID" : "123456789",
    "Country" : "IND",
    "Province" : "XXX",
    "City" : "YYY",
  }
}
```

```

        "HomeAddress" : "ZZZ Road",
        "Email" : "user@example.com",
        "PhoneNo" : "13xxxxx5678",
    }
}

```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
token	uint	R	Obtained search token	23443U
totalCount	int	R	Total number of qualified results	3333

Response Example

```
{
    "token": 23443U,
    "totalCount": 3333
}
```

10.7.9 Obtaining Vehicle Search Results

Request URL	http://<server>/cgi-bin/api/VehicleRegisterDB/doFind		
Method	POST		

Request Params (JSON format in body)

Name	Type	R/O	Description	Example
condition	object	R	Input parameter of face search	
+token	uint	R	Search token	23443U
+beginNumber	uint32	R	Start number of the search. The search starts from the "beginNumber" records, and returns predefined number (count) of records. 0<=beginNumber<= totalCount-1	0
+count	int	R	Number of entries obtained each time	20

Request Example

```
{
    "condition": {
        "token": 23443U,
        "beginNumber": 0,
        "count": 20
    }
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
results	object	R	Output parameter of the search	
+found	int	R	Number of searched entries. One vehicle can be registered to multiple databases.	12

+candidates	object[]	O	Information list of vehicles to be selected	
++Vehicle	object	R	Vehicle information	Vehicle
++DifferentAttributres	char[16][16]	R	A collection of attributes that do not match the database. The license plate must be the same, but other elements might be different for false-registered vehicle. Country: License plate registered place Brand: Vehicle logo Type: Vehicle type VehicleColor: Vehicle color PlateColor: Plate color	["VehicleColor", "Type"]

Response Example

```
{
  "results": {
    "found": 12,
    "candidates": [
      {
        "Vehicle": {Vehicle
        },
        "DifferentAttributres": ["VehicleColor", "Type"]
      },...{}
    ]
  }
}
```

10.7.10 Stopping Searching for Vehicles

Request URL	http://<server>/cgi-bin/api/VehicleRegisterDB/stopFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
token	uint	R	Search token	23443U
Request Example				
{	<pre>"token": 23443U</pre>			

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				
{}				

11.Thermography and Radiometry API

11.1 Thermography Manager

11.1.1 Get Capability of Thermography

Table 11-1

Syntax	http://<server>/cgi-bin/ThermographyManager.cgi?action=getCaps&channel=< ChannelNo >
Method	GET
Description	Get thermography capability.
Example	http://192.168.1.250/cgi-bin/ThermographyManager.cgi?action=getCaps&channel=1
Success Return	<pre>caps.PresetModes = Indoor caps.Brightness.Max = 100 caps.Brightness.Min = 0 caps.Brightness.Step = 1 caps.Sharpness.Max= 100 caps.Sharpness.Min = 0 caps.Sharpness.Step = 5 caps.EZoom.Max= 24 caps.EZoom.Min = 0 caps.EZoom.Step = 1 caps.ThermographyGamma.Max= 8 caps.ThermographyGamma.Min = -8 caps.ThermographyGamma.Step = 1 caps.SmartOptimizer.Max= 100 caps.SmartOptimizer.Min = 0 caps.SmartOptimizer.Step = 5 caps.Agc.Max= 255 caps.Agc.Min = 0 caps.Agc.Step = 5 caps.AgcMaxGain.Max= 255 caps.AgcMaxGain.Min = 0 caps.AgcMaxGain.Step = 5 caps.AgcPlateau.Max= 100 caps.AgcPlateau.Min = 0 caps.AgcPlateau.Step = 5 caps.PresetColorization[i]= Ironbow2 caps.PresetROIModes[j]= Full Screen</pre>
Comment	Parameters in URL: ChannelNo : integer, video channel index which starts from 1.

	<p>Parameters in Response:</p> <p>PresetModes: the preset mode. Range is { "Indoor", "Outdoor", "Default" }</p> <p>PresetColorization: Preset colorization mode. Range is { "WhiteHot", "BlackHot", "Fusion", "Rainbow", "Globow", "Ironbow1", "Ironbow2", "Sepia", "Color1", "Color2", "Icefire", "Rain", "RedHot", "GreenHot" }.</p> <p>PresetROIModes: Preset ROI mode. Range is {"Full Screen", "Sky", "Ground", "Horizon", "Center 75%", "Center 50%", "Center 25%", "Custom"}</p>
--	--

11.1.2 [Config] Thermography Options

- Get thermography options config

Table 11-2

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=ThermographyOptions
Method	GET
Description	Thermography options contain EZoom, Colorization, and SmartOptimizer and so on.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=ThermographyOptions
Success Return	<pre>head.EZoom=0 head.Colorization=White Hot head.SmartOptimizer=10 head.OptimizedRegion.Type=Custom head.OptimizedRegion.Enable= true head.OptimizedRegion.Regions[0][0u]=0 head.OptimizedRegion.Regions[0][1u]=0 head.OptimizedRegion.Regions[0][2u]=0 head.OptimizedRegion.Regions[0][3u]=0 head.Agc=10 head.AgcMaxGain=10 head.AgcPlateau=10 head.Mode="HighTemperature" head.Auto.LowToHigh=13 head.Auto.LHROI=15 head.Auto.HighToLow=12 head.Auto.HLROI=95</pre>
Comment	<p>Parameters in Response:</p> <p>head = table.ThermographyOptions [ChannelNo][0]</p> <p>ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p> <p>Regions: the region is a rectangle</p> <p>i: the array index starts from 0.</p>

- Set thermography options config

Table 11-3

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set thermography options.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&ThermographyOptions[0][0].OptimizedRegion.Type=Gound
Success Return	OK
Comment	<p>Parameters in URL: The paramName and paramValue are in the table below.</p> <p>In table below, head = ThermographyOptions[ChannelNo][0] ChannelNo: integer, array index starts from 0, which means video channel>equals to video channel index -1, and so 0 means channel 1). <i>i</i>: the array index starts from 0.</p>

Appendix

ParamName	ParamValue type	Description
head .EZoom	integer	Range is [0—24]. Range and step are got from interface in "11.1.1 Get Capability of Thermography".
head .Colorization	String	Range is {"White Hot", "Black Hot", "Ironbow2", "IceFire"...}. Range and step are got from interface in "11.1.1 Get Capability of Thermography".
head .SmartOptimizer	integer	Range is [0—100]. Range and step are got from interface in "11.1.1 Get Capability of Thermography".
head .OptimizedRegion.Type	String	Range is {"Full Screen", "Sky", "Ground", "Horizontal", "Center 75%", "Center 50%", "Center 25%", "Custom"}.
head .OptimizedRegion.Enable	bool	true: enable false: not enable
head .OptimizedRegion.Regions[<i>i</i>][0u]	integer	Range is [0—8191]. <i>i</i> : the region index, starts from 0.
head .OptimizedRegion.Regions[<i>i</i>][1u]	integer	Range is [0—8191]. <i>i</i> : the region index, starts from 0.
head .OptimizedRegion.Regions[<i>i</i>][2u]	integer	Range is [0—8191]. <i>i</i> : the region index, starts from 0.
head .OptimizedRegion.Regions[<i>i</i>][3u]	integer	Range is [0—8191]. <i>i</i> : the region index, starts from 0.
head .Agc	integer	Range is [0—255]. Range and step are got from interface in "11.1.1 Get Capability of Thermography".
head .AgcMaxGain	integer	Range is [0—255].

ParamName	ParamValue type	Description
		Range and step are got from interface in "11.1.1 Get Capability of Thermography".
head .AgcPlateau	integer	Range and step are got from interface in "11.1.1 Get Capability of Thermography".
head .Mode	string	Range is {"HighTemperature", "LowTemperature", "Auto"}.
head .Auto.LowToHigh	integer	UInt32
head .Auto.LHROI	integer	UInt32, percentage range is[0—100]
head .Auto.HighToLow	integer	UInt32
head .Auto.HLROI	integer	UInt32, percentage range is[0—100]

11.1.3 Get ExternSystem Information

Table 11-4

Syntax	http://<server>/cgi-bin/ThermographyManager.cgi?action=getExternSystemInfo&channel=< ChannelNo >
Method	GET
Description	Get extern system Info.
Example	http://192.168.1.250/cgi-bin/ThermographyManager.cgi?action=getExternSystemInfo&channel=1
Success Return	sysInfo.SerialNumber = 11111111123 sysInfo.SoftwareVersion = 22222222222222 sysInfo.FirmwareVersion= 33333333333333 sysInfo.LibVersion = 4444444444
Comment	Parameters in URL: ChannelNo : integer, video channel index which starts from 1.

11.1.4 Get Information of Preset Mode

Table 11-5

Syntax	http://<server>/cgi-bin/ThermographyManager.cgi?action=getPresetParam&channel=< ChannelNo >&mode=< modeType >
Method	GET
Description	Get preset mode info.
Example	http://192.168.1.250/cgi-bin/ThermographyManager.cgi?action=getPresetParam&channel=1&mode=Default
Success Return	presetInfo.Brightness = 50 presetInfo.Sharpness= 50 presetInfo.EZoom= 12 presetInfo.ThermographyGamma= 0 presetInfo.Colorization= "White Hot" presetInfo.SmartOptimizer= 10 presetInfo.OptimizedRegion.Type= Full Screen

	<pre> presetInfo.OptimizedRegion.Enable= Full Screen presetInfo.OptimizedRegion.Regions[i][0u]=0 presetInfo.OptimizedRegion.Regions[i][1u]=0 presetInfo.OptimizedRegion.Regions[i][2u]=0 presetInfo.OptimizedRegion.Regions[i][3u]=0 presetInfo.Agc= 10 presetInfo.AgcMaxGain=10 presetInfo.AgcPlateau = 10 </pre>
Comment	<p>Parameters in URL:</p> <p>ChannelNo: integer, video channel index which starts from 1.</p> <p>modeType: depends on capability , get from interface in getCaps</p> <p>Parameters in Response :</p> <p>Regions : the region is a rectangle</p> <p>i : the array index.</p>

11.1.5 Get Optimized Region Information

Table 11-6

Syntax	<code>http://<server>/cgi-bin/ThermographyManager.cgi?action=getOptimizedRegion&channel=<ChannelNo></code>
Method	GET
Description	Get optimized region info.
Example	<code>http://192.168.1.250/cgi-bin/ThermographyManager.cgi?action=getOptimizedRegion&channel=1</code>
Success Return	<pre> optimizedRegion.Type= Full Screen optimizedRegion.Enable= true optimizedRegion.Regions[i][0u]=0 optimizedRegion.Regions[i][1u]=0 optimizedRegion.Regions[i][2u]=0 optimizedRegion.Regions[i][3u]=0 </pre>
Comment	<p>Parameters in URL:</p> <p>ChannelNo: integer, video channel index which starts from 1.</p> <p>Parameters in Response:</p> <p>Regions : the region is a rectangle</p> <p>i: the region index.</p>

11.1.6 Enable Shutter

Table 11-7

Syntax	<code>http://<server>/cgi-bin/ThermographyManager.cgi?action=enableShutter&channel=<ChannelNo>&enable=<Enable></code>
Method	GET
Description	Shutter control, whether enable shutter.

Example	<code>http://192.168.1.250/cgi-bin/ThermographyManager.cgi?action=enableShutter&channel=1&enable=true</code>
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>ChannelNo: integer, video channel index which starts from 1.</p> <p>Enable: true or false, enable or not.</p>

11.1.7 Fix Focus

Table 11-8

Syntax	<code>http://<server>/cgi-bin/ThermographyManager.cgi?action=fixFocus&linkVideoChannel[0]=<ChannelNo>&linkVideoChannel[1]=<ChannelNo> [&speed=<SpeedValue>]</code>
Method	GET
Description	The visual channel change focus to the same as the thermography channel.
Example	<code>http://192.168.1.250/cgi-bin/ThermographyManager.cgi?action=fixFocus&linkVideoChannel[0]=1&linkVideoChannel[1]=2</code>
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>ChannelNo: integer, video channel index which starts from 1.</p> <p>SpeedValue: float, range is 0.0-1.0.</p>

11.1.8 Do Flat Field Correction

Table 11-9

Syntax	<code>http://<server>/cgi-bin/ThermographyManager.cgi?action=doFFC&channel=<ChannelNo></code>
Method	GET
Description	Do flat field correction.
Example	<code>http://192.168.1.250/cgi-bin/ThermographyManager.cgi?action=doFFC&channel=1</code>
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>ChannelNo: integer, video channel index which starts from 1.</p>

11.2 Radiometry

11.2.1 Get Capability of Radiometry

Table 11-10

Syntax	<code>http://<server>/cgi-bin/RadiometryManager.cgi?action=getCaps [&channel=<ChannelNo>]</code>
--------	--

Method	GET
Description	Get the capabilities of radiometry manager.
Example	http://192.168.1.250/cgi-bin/RadiometryManager.cgi?action=getCaps&channel=1
Success Return	caps.TotalNum.MaxNum=8 caps.TotalNum.Spot.MaxSpots=8 caps.TotalNum.Line.MaxLines=1 caps.TotalNum.Area.MaxAreas=8 caps.TemperPresets.MaxPresets=256 caps.MeterInfo.Type[0u]=Spot caps.MeterInfo.Type[1u]=Area caps.MeterInfo.ObjectEmissivity.Max=100 caps.MeterInfo.ObjectEmissivity.Min=0 caps.MeterInfo.ObjectEmissivity.Default=0 caps.MeterInfo.ObjectEmissivity.Step=1 caps.MeterInfo.ObjectDistanceMeter.Max=100 caps.MeterInfo.ObjectDistanceMeter.Min=0 caps.MeterInfo.ObjectDistanceMeter.Default=0 caps.MeterInfo.ObjectDistanceMeter.Step=1 caps.MeterInfo.ReflectedTemperature.Max=100 caps.MeterInfo.ReflectedTemperature.Min=0 caps.MeterInfo.ReflectedTemperature.Default=0 caps.MeterInfo.ReflectedTemperature.Step=1 caps.MeterInfo.RelativeHumidity.Max=100 caps.MeterInfo.RelativeHumidity.Min=0 caps.MeterInfo.RelativeHumidity.Default=0 caps.MeterInfo.RelativeHumidity.Step=1 caps.MeterInfo.AtmosphericTemperature.Max=100 caps.MeterInfo.AtmosphericTemperature.Min=0 caps.MeterInfo.AtmosphericTemperature.Default=0 caps.MeterInfo.AtmosphericTemperature.Step=1 caps.Statistics.MinPeriod=60 caps.Isotherm.MaxTemp=327.0 caps.Isotherm.MinTemp=-20.0
Comment	Parameters in URL: ChannelNo: integer, video channel index which starts from 1.

11.2.2 [Config] Heat Image Thermometry

- Get heat image thermometry config

Table 11-11

Syntax	<a href="http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=HeatImagingThermometry">http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=HeatImagingThermometry
Method	GET
Description	Get HeatImagingThermometry Config.

Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=HeatImagingThermometry
Success Return	table.RelativeHumidity = 50 table.AtmosphericTemperature =20 table.ObjectEmissivity =1 table.ObjectDistance =100 table.ReflectedTemperature=20 table.TemperatureUnit= Centigrade table.Isotherm.Enable=true table.Isotherm.MaxValue=50 table.Isotherm.MinValue=0 table.Isotherm.ColorBarDisplay=true table.HotSpotFollow=true table.TemperEnable=true
Comment	—

- Set heat image thermometry config

Table 11-12

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set HeatImagingThermometry Config
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&HeatImagingThermometry.RelativeHumidity=50&HeatImagingThermometry.ObjectDistance=20
Success Return	OK
Comment	Parameters in URL: The paramName and paramValue are in the table below.

Appendix

ParamName	ParamValue type	Description
HeatImagingThermometry.RelativeHumidity	integer	The Relative Humidity range and step are got from interface in getCaps .
HeatImagingThermometry.AtmosphericTemperature	float	The Atmospheric Temperature range and step are got from interface in getCaps .
HeatImagingThermometry.ObjectEmissivity	float	The Object Emissivity range and step are got from interface in getCaps .
HeatImagingThermometry.ObjectDistance	integer	The Object Distance range and step are got from interface in getCaps . Unit is meter.

ParamName	ParamValue type	Description
HeatImagingThermometry.ReflectedTemperature	float	The Reflected Temperature range and step are got from interface in getCaps
HeatImagingThermometry.TemperatureUnit	string	Range is {Centigrade, Fahrenheit}.
HeatImagingThermometry.Isotherm.Enable	bool	true or false
HeatImagingThermometry.Isotherm.MaxValue	float	MaxValue range is got form interface in getCaps. MaxValue must be bigger than MinValue
HeatImagingThermometry.Isotherm.MinValue	float	MinValue range is got form interface in getCaps. MinValue must be smaller than MaxValue.
HeatImagingThermometry.Isotherm.ColorBarDisplay	bool	true or false
HeatImagingThermometry.HotSpotFollow	bool	true or false
HeatImagingThermometry.TemperEnable	bool	true or false

11.2.3 [Config] Thermometry Rule

- Get thermometry rule config

Table 11-13

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=ThermometryRule
Method	GET
Description	Get thermometry rule.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=Thermometry Rule
Success Return	<pre> head.Enable = true head.PresetId =0 head.RuleId=0 head.Name=SpotName head.Type=Spot head.MeterRegion.Coordinates[PointNo][0]= 0 head.MeterRegion.Coordinates[PointNo][1]= 0 ... head.T=3 head.Alarm.Id=0 head.Alarm.Enable=true head.Alarm.Result =Max head.Alarm.AlarmCondition=Below head.Alarm.Threshold=20.0 head.Alarm.PreThreshold=10.0 head.Alarm.PreDuration=30 head.Alarm.Hysteresis=0.1 </pre>

	<p>head.Alarm.Duration=30 head.LocalParameters.Enable=true head.LocalParameters.ObjectEmissivity=0.95 head.LocalParameters.ObjectDistance=0.95 head.LocalParameters.ReflectedTemp=0</p>
Comment	<p>Parameters in Response :</p> <p>head =table.ThermometryRule[ChannelNo][RuleNo]</p> <p>ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p> <p>RuleNo =rule index.</p> <p>PointNo = point index</p> <p>Alarm= AlarmSetting[AlarmNo]</p> <p>AlarmNo = alarm index</p>

- Set thermometry rule config

Table 11-14

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&< paramName >=< paramValue >[&< paramName >=< paramValue >...]
Method	GET
Description	Set thermometry rule.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&ThermometryRule[0][0].Name=name1
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>The paramName and paramValue are in the table below.</p> <p>In table below,</p> <p>head = ThermometryRule[ChannelNo][RuleNo]</p> <p>ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p> <p>PointNo = point index</p> <p>RuleNo =rule index.</p> <p>Alarm= AlarmSetting[AlarmNo]</p> <p>AlarmNo = alarm index</p>

Appendix

ParamName	ParamValue type	Description
head.Enable	bool	Enable/Disable
head.PresetId	integer	Range [0—PresetMax] PresetMax is got from interface in <u>GetCurrentProtocolCaps</u> .
head.RuleId	integer	Range [0—MaxNum] MaxNum is got from interface in <u>getCaps</u> .

ParamName	ParamValue type	Description
head.Name	string	Radiometry rule name. char[64]
head.Type	string	Range is {Spot, Line, Area }.
head.MeterRegion.Coordinates[PointNo][0]	integer	Range [0—8091] The Xscale of Region/Line point
head.MeterRegion.Coordinates[PointNo][1]	integer	Range [0—8091] The Yscale of Region/Line point
head.T	integer	Temperature Sample period. Unit is Second.
head.Alarm.Id	integer	Range [0 — 65535],unique alarm id
head.Alarm.Enable	bool	Enable/Disable
head.Alarm.Result	string	Depend on the value of Type Spot : {Value} Line: { Max, Min, Aver} Area: {Max, Min, Aver, Std, Mid, ISO}
head.Alarm.AlarmCondition	string	Range is {Below, Match , Above }
head.Alarm.Threshold	float	Alarm threshold
head.Alarm.Hysteresis	float	Alarm hysteresis
head.Alarm.Duration	integer	The duration time of alarm. Unit is second
Head.Alarm.PreThreshold	float	PreAlarm threshold
head.Alarm.PreDuration	integer	The duration time of preAlarm. Unit is second
head.LocalParameters.Enable	bool	Enable/Disable
head.LocalParameters.ObjectEmissivity	float	Range [0 — 1] Accuracy is 0.01
head.LocalParameters.ObjectDistance	float	Object distance The range is got from interface in <u>getCaps</u> .
head.LocalParameters.ReflectedTemp	float	Object Reflected Temperature The range is got from interface in <u>getCaps</u> .

11.2.4 [Config] Heat Image Temper Event

- Get heat image temper event config

Table 11-15

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=HeatImagingTemper
--------	---

Method	GET
Description	Get Heat Imaging Temper config
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=HeatImagingTemper
Success Return	<p>head.Enable=false</p> <p>head.EventHandler.paramName = paramValue</p>

- Set heat image temper event config

Table 11-16

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set Heat Imaging Temper config
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&HeatImagingTemper[0].Enable=false&HeatImagingTemper[0].EventHandler.BeepEnable=false
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>The paramName and paramValue are in the table below.</p> <p>In table below,</p> <p>head= HeatImagingTemper[<i>ChannelNo</i>]</p> <p>ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p>

Appendix

ParamName	ParamValue type	Description
head.Enable	bool	Enable/Disable Heat Imaging Temper feature.
head.EventHandler	—	Setting of EventHandler is described in SetEventHandler .

11.2.5 Get Temperature of Particular Point

Table 11-17

Syntax	http://<server>/cgi-bin/RadiometryManager.cgi?action=getRandomPointTemper&channel=<ChannelNo>&coordinate[0]=x &coordinate[1]=y
Method	GET
Description	Get temperature values of random point.
Example	http://192.168.1.250/cgi-bin/RadiometryManager.cgi?action=getRandomPointTemper&channel=1&coordinate[0]=1024&coordinate[1]=1024

Success Return	TemplInfo.Type=Spot TemplInfo.TemperAver=27.5
Comment	Parameters in URL: ChannelNo : integer, video channel index which starts from 1. x : The Xscale of the point y : The Yscale of the point

11.2.6 Get Temperature of Particular Condition

Table 11-18

Syntax	http://<server>/cgi-bin/RadiometryManager.cgi?action=getTemper&< paramName >=< paramValue >[&< paramName >=< paramValue >...]
Method	GET
Description	Get temperature values from rules which have been set.
Example	http://192.168.1.250/cgi-bin/RadiometryManager.cgi?action=getTemper&condition.PresetId=0&condition.RuleId=0&condition.Type=Spot&condition.Name=Spot1&condition.channel=1
Success Return	TemplInfo.Type=Spot TemplInfo.TemperAver=27.5
Comment	Parameters in URL: The paramName and paramValue are in the table below.

Appendix

ParamName	ParamValue type	Description
condition.Channel	integer	Video channel index. Start from 1
condition.PresetId	integer	Range [0- PresetMax] PresetMax is got from interface in GetCurrentProtocolCaps .
condition.RuleId	integer	Range [0- MaxNum] MaxNum is got from interface in getCaps .
condition.Type	string	Range is {Spot, Line, Area}.
condition.Name	string	Name is got from interface in GetThermometryRuleConfig .

11.2.7 Find Temperature Information

- Start to query temperature information

Table 11-19

Syntax	http://<server>/cgi-bin/RadiometryManager.cgi?action=startFind&condition.StartTime=< StartTimeValue >&condition.EndTime=< EndTimeValue >&condition.Type=< TypeValue >&condition.channel=< ChannelValue >&condition.Period=< PeriodValue >
Method	GET
Description	Start to query the history data of temperature values.
Example	http://192.168.1.250/cgi-bin/RadiometryManager.cgi?action=startFind&condition.StartTi

	me=2010-04-01%200:00:00&condition.EndTime=2010-04-08%200:00:00&condition.Type=Spot&condition.channel=1&condition.Period=5
Success Return	token=46878 totalCount=333
Comment	The parameters in bold face are as table below.

Appendix

ParamName	ParamValue type	Description
condition.StartTime	string	The start time to find.
condition.EndTime	string	The end time to find.
condition.Type	string	The type of data. Range is {Spot, Line, Area}
condition.channel	integer	Video channel index. Start from 1
condition.Period	integer	Range is {5, 10, 15, 30}, minute

2. Get the data of temperature

Table 11-20

Syntax	http://<server>/cgi-bin/RadiometryManager.cgi?action=doFind& token =<tokenvalue>& beginNumber =<BeginNumber>& count =<findNum>
Method	GET
Description	Get the history data of temperature.
Example	http://192.168.1.250/cgi-bin/RadiometryManager.cgi?action=doFind&token=46878&be ginNumber=16&count=16
Success Return	found=12 info[i].Time=2010-04-08 16:12:46 info[i].PresetId=0 info[i].RuleId=0 info[i].Type=Spot info[i].Name=xxxx info[i].Coordinate[0]=1024 info[i].Coordinate[1]=2048 info[i].Channel=0 info[i].TemperatureUnit=Centigrade info[i].QueryTemperInfo.TemperAve=50.1 info[i].QueryTemperInfo.TemperMax=50.2 info[i].QueryTemperInfo.TemperMin=50.0
Comment	<p>Parameters in URL:</p> <p>token: query token, get from interface of the first step above.</p> <p>beginNumber: the begin index in this query.</p> <p>count: the number you want to query.</p> <p>Params in Resp:</p> <p><i>i</i>: the array index.</p>

3. Stop finding temperature information

Table 11-21

Syntax	http://<server>/cgi-bin/RadiometryManager.cgi?action=stopFind& token =<tokenvalue>
--------	---

Method	GET
Description	Stop to find the history data of temperature values.
Example	http://192.168.1.250/cgi-bin/RadiometryManager.cgi?action=stopFind&token=46878
Success Return	OK

Comment **token**: query token, get from interface of the first step.

11.2.8 Subscribe Temperature Information

Subscribe temperature information

Request URL	http://<server>/cgi-bin/RadiometryManager.cgi?action=attachTemper		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	R	Video channel number, starting from 1
heartbeat	int	O	Heartbeat interval. The unit is second, and the default value is 5. When the device sends temperature data in the response, it will periodically send a heartbeat message to keep it alive according to the heartbeat interval. The content of the message is the string "Heartbeat".
Request Example			
http://192.168.1.250/cgi-bin/RadiometryManager.cgi?action=attachTemper&channel=2&heartbeat=5			

Response Params (multipart ; key=value format in body)				
Name	Type	R/O	Description	Example
info	object[]	R	Temperature information array	
+Time	char[32]	R	Record time, format: yyyy-MM-dd HH:mm:ss	"2010-04-08 16:12:46"
+PresetId	int	O	Preset number, starting from 1	1
+RuleId	int	O	Rule number	1
+Type	char[16]	R	Search Type "Spot": Spot "Area": Area "Line": Line	"Spot"
+Name	char[64]	O	Query item name	"xxxx"
+Coordinate	int[8][2]	O	Query the coordinates of temperature monitoring point, up to 8, using the 8192 relative coordinate system.	[[1024, 2048]]
+Channel	int	O	Video channel number, starting from 0	0
+TemperatureUnit	char[16]	R	Temperature unit, value: "Centigrade" : Centigrade "Fahrenheit": Fahrenheit	"Centigrade"
+QueryTemperInfo	object	R	Temperature information	
++TemperAve	float	R	Average temperature	50.1

++TemperMax	float	R	Maximum temperature	50.2
++TemperMin	float	R	Minimum temperature	50.0

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: closed

--<boundary>

Content-Type: text/plain

Content-Length: <data length>

info[0].Time=2010-04-08 16:12:46

info[0].PresetId=1

info[0].RuleId=1

info[0].Type=Spot

info[0].Name=xxxx

info[0].Coordinate[0][0]=1024

info[0].Coordinate[0][1]=2048

info[0].Channel=0

info[0].TemperatureUnit=Centigrade

info[0].QueryTemperInfo.TemperAve=50.1

info[0].QueryTemperInfo.TemperMax=50.2

info[0].QueryTemperInfo.TemperMin=50.0

.....

--<boundary>

Content-Type: text/plain

Content-Length: 11

Heartbeat

--<boundary>

.....

11.2.9 Subscribe Radiometry Data

Subscribe temperature distribution data

Request URL	http://<server>/cgi-bin/RadiometryManager.cgi?action=attachProc			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	Video channel number, starting from 1	1
heartbeat	int	O	Heartbeat interval. The unit is second, the value range is [1,60], the default value is 5. When the device sends temperature data in the response, it will periodically send a heartbeat message to keep it alive according to the heartbeat interval. The content of the	5

			message is the string "Heartbeat".	
Request Example				
http://192.168.1.250/cgi-bin/RadiometryManager.cgi?action=attachTemper&channel=2&heartbeat=5				
Response Params (multipart ; key=value format in body)				
Name	Type	R/O	Description	Example
dataInfo	object	R	Temperature distribution data information	
+Width	int	R	Image width, in pixels	1920
+Height	int	R	Image height, in pixels	1080
+Channel	int	O	Video channel number, starting from 0	0
+Time	char[32]	R	Fetching data time, format: yyyy-MM-dd HH:mm:ss	"2010-04-08 16:12:46"
+Length	int	O	Data length	2073600
+sensorType	char[16]	O	Algorithm type, value: "Tau"	"Tau"
+Unzip	object	O	Decompression parameter, used when sensorType is Tau	
++ParamR	uint	O	Parameter R required for temperature conversion	1
++ParamB	uint	O	Parameter B required for temperature conversion	1
++ParamF	uint	O	Parameter F required for temperature conversion	1
++ParamO	uint	O	Parameter O required for temperature conversion	1
Response Example				
HTTP/1.1 200 OK				
Server: Device/1.0				
Content-Type: multipart/x-mixed-replace; boundary=<boundary>				
Connection: closed				
--<boundary>				
Content-Type: text/plain				
Content-Length: <data length>				
dataInfo.Width=1920				
dataInfo.Height=1080				
dataInfo.Channel=0				
dataInfo.Time=2010-05-25 00:00:00				
dataInfo.Length=2073600				
dataInfo.sensorType="Tau"				
dataInfo.Unzip.ParamR=1				
dataInfo.Unzip.ParamB=1				
dataInfo.Unzip.ParamF=1				
dataInfo.Unzip.ParamO=1				
--<boundary>				
Content-Type: application/octet-stream				

Content-Length: <data length>

<Binary data>

--<boundary>

Content-Type: text/plain

Content-Length: 11

Heartbeat

--<boundary>

.....

11.2.10 Fetch Radiometry Data

Table 11-22

Syntax	http://<server>/cgi-bin/RadiometryManager.cgi?action=toFetch&channel=< ChannelNo >
Method	GET
Description	Start to fetch radiometry data.
Example	http://192.168.1.250/cgi-bin/RadiometryManager.cgi?action=toFetch&channel=2
Success Return	status=Ready
Comment	ChannelNo : integer, video channel index which starts from 1. status : Range is {Ready, Busy}. "Ready"means service available and "Busy"means service busy.

11.2.11 [Config] Get FireWarning Config

Table 11-23

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=FireWarning
Method	GET
Description	Get FireWarning config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=FireWarning
Success Return	head.Enable=true head.PresetId=0 head.Row = 31 head.Col = 40 head.Mode="Auto" head.SmdFilterEnable=false head.MovingTargetFilterEnable=false head.SunReflectEnable=false head.TimeDurationEnable=false head.FireDuration= 15 head.DetectWindow[windowsNum].Regions[0]=123468789 head.DetectWindow[windowsNum].Regions[1]=123468789 head.DetectWindow[windowsNum].Regions[2]=123468789

	head.DetectWindow[windowsNum].Regions[3]=123468789 head.DetectWindow[windowsNum].Postion[0]=0 head.DetectWindow[windowsNum].Postion[1]=0 head.DetectWindow[windowsNum].Postion[2]=0 head.DetectWindow[windowsNum].Postion[3]=0 head.DetectWindow[windowsNum].Sensitivity = 95 head.DetectWindow[windowsNum].Id=1 head.DetectWindow[windowsNum].Name="windName" head.EventHandler=(output of EventHandler is described in GetEventHandler)
Comment	<p>Parameters in Response:</p> <p>head= table.FireWarning[ChnkleNo][RuleNum]</p> <p>ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p> <p>RuleNum: integer, array index starts from 0, which means rules in each Preset, If FireWarningMode is "SpaceExClude", then only the first rule used.</p> <p>windowsNum: support 4 Nums if FireWarningMode is "PtzPreset", 8 if "SpaceExClude"</p> <p>Postion: only valid in "SpaceExClude"Mode</p>

11.2.12 [Config] Set FireWarning Config

Table 11-24

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]
Method	GET
Description	Set FireWarning config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&FireWarning[0][0].Enable=false
Success Return	OK
Comment	head = FireWarning[ChNum][RuleNum]

Appendix:

ParamName	ParamValue type	Description
head .Enable	bool	whether fire detect take effect
head .PresetId	integer	The PresetId
head .Row	integer	rows of fire detect area
head .Col	integer	cols of fire detect area
head .Mode	string	Range is {"Auto","Normal"}.
Head .SmdFilterEnable	bool	whether smd filter take effect
Head .MovingTargetFilterEnable	bool	whether moving target filter take effect
Head .SunReflectEnable	bool	whether sun reflect filter take effect
head .TimeDurationEnable	bool	whether include fire detect duration,only take effect in

ParamName	ParamValue type	Description
		SpaceExClude mode
head .FireDuration	integer	fire last times
head .DetectWindow[windowsNum].Regions[0]	integer	detect area mask
head .DetectWindow[windowsNum].Sensitivity	integer	Rage {0,100}
head .DetectWindow[windowsNum].Id	integer	Rage {0,...}
head .DetectWindow[windowsNum].Name	string	detect window name
head .EventHandler	EventHandler	Setting of EventHandler is described in SetEventHandler.

11.2.13 [Config] Get FireWarningMode Config

Table 11-25

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=FireWarningMode
Method	GET
Description	Get FireWarningMode config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=FireWarningMode
Success Return	head.Mode="PtzPreset"
Comment	Parameters in Response: head= table.FireWarningMode[ChannelNo] <i>ChannelNo:</i> integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

11.2.14 [Config] Set FireWarningMode Config

Table 11-26

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue> [&<paramName>=<paramValue> ...]
Method	GET
Description	Set FireWarningMode config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&FireWarningMode[0].Mode ="SpaceExClude"
Success Return	OK
Comment	head = FireWarningMode[ChannelNo]

Appendix:

ParamName	ParamValue type	Description
head .Mode	string	Rage is {"PtzPreset", "SpaceExClude"}

11.2.15 Get Current Hot Cold Spot

Table 11-27

URL	http://<server>/cgi-bin/TemperCorrection.cgi?action=getCurrentHotColdSpot		
Method	GET		
Description	Get the max and min temperature values.		
[Request Params] (key=value format in URL)			
channel	int	R	video channel index which starts from 1.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
info	object	R	The current hot code spot info.
+HotPoint	array< int >	R	The hot spot position, must be two int, means x and y value, coordinate remap to 0 — 8192.
+HotSpotValue	double	R	The hot spot temperature value.
+ColdPoint	array< int >	O	The cold spot position, must be two int, means x and y value, coordinate remap to 0 — 8192.
+ColdSpotValue	double	O	The cold spot temperature value.
+TemperatureUnit	int	O	The temperature unit : 0 Centigrade, 1 Fahrenheit
[Example]			
Request	GET http://192.168.1.250/cgi-bin/TemperCorrection.cgi?action=getCurrentHotColdSpot &channel=1		
Response	info.HotPoint[0]=1150 info.HotPoint[1]=2320 info.HotSpotValue=35.5 info.ColdPoint[0]=5452 info.ColdPoint[1]=6192 info.ColdSpotValue=24.3 info.TemperatureUnit=0		

11.2.16 [Config] Heat Image Temper PreAlarm Event

- Get heat image temper pre event config

Table 11-28

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=PreAlarmEvent
Method	GET
Description	Get Heat Imaging Temper PreAlarm Event config
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=PreAlarmEvent
Success Return	head.Enable=false head.EventHandler. paramName = paramValue
Comment	Parameters in Response: head= table.PreAlarmEvent[ChannelNo] ChannelNo: integer, array index starts from 0, which means video channel (equals to

video channel index -1, and so 0 means channel 1).

- Set heat image temper pre event config

Table 11-29

Syntax	<code>http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]</code>
Method	GET
Description	Set Heat Imaging Temper PreAlarm Event config
Example	<code>http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&PreAlarmEvent[0].Enable=false&PreAlarmEvent[0].EventHandler.BeepEnable=false</code>
Success Return	OK
Comment	<p>Parameters in URL: The paramName and paramValue are in the table below.</p> <p>In table below, head=PreAlarmEvent[ChannelNo] ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p>

Appendix

ParamName	ParamValue type	Description
head .Enable	bool	Enable/Disable Heat Imaging Temper feature.
head .EventHandler	—	Setting of EventHandler is described in SetEventHandler .

11.2.17 Get Heat Map Info

Get Heat Map Info.

Request URL	<code>http://<server>/cgi-bin/RadiometryManager.cgi?action=getHeatMapsDirectly&channel=<ChannelNo></code>			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	video channel index which starts from 1	2
Request Example				<code>http://192.168.1.250/cgi-bin/RadiometryManager.cgi?action=getHeatMapsDirectly&channel=2</code>

Response Params	(multipart, binary in body)			
Name	Type	R/O	Description	Example
dataInfo	object	O	Data info	
+Height	integer	O	height	0
+Width	integer	O	width	0

+Channel	integer	O	Channel number	0
+Time	string	O	Time of getting data	2010-05-25 00:00:00
+Length	integer	O	Data length	0
+sensorType	string	O	Algorithm type	"Tau"
+Unzip	object	O	unzip parameters, it works when sensorType was "Tau".	
++ParamR	integer	O	Parameter R for temperature conversion	1
++ParamB	integer	O	Parameter B for temperature conversion	1
++ParamF	integer	O	Parameter F for temperature conversion	1
++ParamO	integer	O	Parameter O for temperature conversion	1

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: closed

--<boundary>

Content-Type: text/plain

Content-Length: <data length>

dataInfo.Height=0

dataInfo.Width=0

dataInfo.Channel=0

dataInfo.Time=2010-05-25 00:00:00

dataInfo.Length=0

dataInfo.sensorType="Tau"

dataInfo.Unzip.ParamR=1

dataInfo.Unzip.ParamB=1

dataInfo.Unzip.ParamF=1

dataInfo.Unzip.ParamO=1

--<boundary>

Content-Type: application/octet-stream

Content-Length: <data length>

<Binary data>

--<boundary>

.....

11.3 TemperCustom

11.3.1 Set Environment Temperature

Syntax	http://<server>/cgi-bin/TemperCustom.cgi?action=setEnvTemp		
Method	GET		
Description	Set environment temperature		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
EnvironmentTemp	int	R	environment temperature Magnified 100 times,the unit is 0.01Celsius degree
[Response Params] (OK)			
Name	Type	R/O	Param Description
[Example]			
Request	http://192.168.1.250/cgi-bin/TemperCustom.cgi?action=setEnvTemp&EnvironmentTemp=2800		
Response	OK		

12. Access Control API

12.1 Access Control

12.1.1 Open Door

Open the door

Request URL	http://<server>/cgi-bin/accessControl.cgi?action=openDoor		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	R	the index of door, starts from 1
UserID	int	O	remote user ID
Type	char[16]	O	the open type, default value is "Remote"
Request Example			
http://192.168.1.250/cgi-bin/accessControl.cgi?action=openDoor&channel=1&UserID=101&Type=Remote			

Response Params (OK in body)

Response Example

OK

12.1.2 Close Door

Close the door

Request URL	http://<server>/cgi-bin/accessControl.cgi?action=closeDoor		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	R	the index of door, starts from 1.
UserID	int	O	remote user ID.
Type	char[16]	O	the open type, default value is "Remote".
Request Example			
http://192.168.1.250/cgi-bin/accessControl.cgi?action=closeDoor&channel=1&UserID=101&Type=Remote			

Response Params (OK in body)

Response Example

OK

12.1.3 Get Door Status

Get status of the door

Request URL	http://<server>/cgi-bin/accessControl.cgi?action=getDoorStatus		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	R	the index of door, starts from 1
Request Example			
http://192.168.1.250/cgi-bin/accessControl.cgi?action=getDoorStatus&channel=1			

Response Params (key=value format in body)			
Name	Type	R/O	Description
Info	object	R	door status info
+status	char[16]	R	door status, the range is {Open, Break, Close}
Response Example			
Info.status=Open			

12.1.4 Get Lock Status

Syntax	http://<server>/cgi-bin/accessControl.cgi?action=getLockStatus&channel=<ChannelNo>
Method	GET
Description	Get status of the lock.
Example	http://192.168.1.250/cgi-bin/accessControl.cgi?action= getLockStatus &channel=1
Success	onLineStatus = "Online"
Return	status = "Close"
Comment	Parameters in URL : ChannelNo : integer, the index of lock, starts from 1. Default is 1 if not present. Parameters in Response : status : the range is {"Open", "Close", "Abnormal", "FakeLocked", "Unknown"} onLineStatus : the range is { "OnLine", "OffLine" }

12.1.5 Capture Fingerprint

Only ID verification terminals such as ASHZ320/520 are supported. Common access control is not supported.

Table 12-1

Syntax	http://<server>/cgi-bin/accessControl.cgi?action=captureFingerprint		
Method	GET		
Description	Capture fingerprint data, then report it via 'Fingerprint' event.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
info	object	R	fingerprint data
+ReaderID	string	R	Reader machine ID

+FingerPrintName	string	O	Fingerprint name (For smart building products)
+ UserID	string	O	User Identity (For smart building products)
heartbeat	int	R	Heartbeat interval (sec)
timeout	int	R	Timeout (sec)
[Response Params] (multipart , json format in body , Heartbeat in body)			
[Example]			
Request	http://<server>/cgi-bin/accessControl.cgi?action=captureFingerprint&info.ReaderID=101&info.FingerPrintName=aaaa&info.UserID=10221&heartbeat=5&timeout=10		
Response	<p>HTTP/1.1 200 OK Cache-Control: no-cache Pragma: no-cache Expires: Thu, 01 Dec 2099 16:00:00 GMT Connection: close Content-Type: multipart/x-mixed-replace; boundary=myboundary</p> <p>--myboundary Content-Type: text/plain Content-Length: 238002</p> <pre>Code=Fingerprint;action=Pulse;index=0;data={ "CollectResult": true, "FingerPrintID": 60, "ReINDo": 1234, "ReaderID": "1", "UserID": "001", "CardNo": "001", "FingerprintPacket": { "Length": 512, "Count": 3 }, "FingerprintData": "xxx", "BinaryData": Binary Data }</pre> <p>--myboundary Content-Type: text/plain Content-Length: 9</p> <p>Heartbeat</p> <p>--myboundary</p> <p>.....</p>		

12.1.6 Capture Face Picture

Only ID verification terminals such as ASHZ320/520 are supported. Common access control is not supported.

Table 12-2

URL	http://<server>/cgi-bin/accessControl.cgi?action=captureCmd		
Method	GET		
Description	Capture face data, then report it via 'CitizenPictureCompare' event.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
+type	int	R	Capture entity types: 0x01 - Face 0x02 - Identification card
+ UserID	string	O	User ID
heartbeat	int	R	Heartbeat interval (sec)
timeout	int	R	Timeout (sec)
[Response Params] (multipart , json format in body , Heartbeat in body)			
[Example]			
Request	http://<server>/cgi-bin/accessControl.cgi?action=captureCmd&type=1&UserID=10221&heartbeat=5&timeout=10		
Response	<p>HTTP/1.1 200 OK Cache-Control: no-cache Pragma: no-cache Expires: Thu, 01 Dec 2099 16:00:00 GMT Connection: close Content-Type: multipart/x-mixed-replace; boundary=myboundary</p> <p>--myboundary Content-Type: text/plain Content-Length: 2380</p> <p>Code=CitizenPictureCompare;action=Pulse;index=0;data={ "UTC": 1999999999, "CompareResult": true, "Similarity": 90, "Threshold": 80, "Citizen": "Rohit", "Sex": 1, "Minzu": 1, "Birth": "1980-01-01", ... }</p> <p>--myboundary Content-Type: text/plain Content-Length: 9</p> <p>Heartbeat</p> <p>--myboundary</p>		

12.1.7 Query AccessControl Record

Table 12-3

URL	http://<server>/cgi-bin/recordFinder.cgi?action=find&name=<RecordName>		
Method	GET		
Description	Query AccessControl record.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	The record table name. It should be " AccessControlCardRec ".
count	int	O	Max result to return, default is 1024.
StartTime	string	O	The start of the record's CreateTime.
EndTime	string	O	The end of the record's CreateTime.
condition	object	O	Search condition.
+CardNo	string	O	Access user card number.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
totalCount	int	O	Total record num that find.
found	int	O	Record num that returned.
records	array<object>	R	The records that returned.
+RelNDo	int	R	Record id.
+CreateTime	int	O	Record create time, UTC time.
+CardNo	int	R	Access user card number.
+CardName	string	O	Access user card name, max string length is 31.
+CardType	int	O	The card type: 0: Normal Card, 1: VIP Card, 2: Visitor Card, 3: Patrol Card, 4: Blocklist Card, 5: Stress Card, 0xff: Mother Card
+Password	string	O	The Access card's password.
+UserID	string	R	The user's id.
+Type	string	O	The event type. It can be: " Entry ", " Exit ".
+Status	int	O	Open door result. It can be: 0—failed, 1—success. If this param does not exist, that means success.
+Method	int	R	Open door method. It can be: <ul style="list-style-type: none"> • 0: by password • 1: by access card • 2: by access card and then password • 3: by password and then access card • 6: by fingerprint • 15: by face recognition
+Door	int	O	The index of the door. (This param is not supported by video talk device)

+ReaderID	string	O	The access user card ID of reader. (This param is not supported by video talk device)
+ErrorCode	int	O	The error code, valid only when Status is 0.
+URL	string	O	The picture's URL, max string length is 127. (This param is not supported by access control device)
+RecordURL	string	O	The record video's URL, max string length is 127. (This param is not supported by access control device)
+IsOverTemperature	bool	O	is over temperature
+TemperatureUnit	integer	O	temperature unit:{ 0: centigrade 1: Fahrenheit 2: Kelvin }
+CurrentTemperature	float	O	current temperature
+CitizenIDResult	bool	O	citizen ID compare result
+CitizenIDName	string	O	citizen name
+CitizenIDNo	string	O	citizen ID number
+CitizenIDSex	integer	O	sex:{ 0: unknow 1: male 2: female 9: not stated }
+CitizenIDMinzu	integer	O	citizen ID Minzu
+CitizenIDBirth	string	O	birthday, format as "1980-01-01"
+CitizenIDAddress	string	O	address
+CitizenIDAuthority	string	O	signing and issuing organization
+CitizenIDStart	string	O	Effective date
+CitizenIDEnd	string	O	Expiration date, and "Endless" mens Long term effectiveness
[Example]			
Request	GET http://192.168.1.250/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCard Rec&StartTime=123456700&EndTime=123456800&condition.CardNo=12001&count=100		
Response	totalCount=1000 found=100 records[0].ReINDo=12345 records[0].CreateTime=123456789 records[0].CardNo=12001 records[0].CardName=Rohit records[0].UserID=Rohit records[0].Type=Entry records[0].Method=1		

	<pre> ... records[1].ReINDo=13579 records[1].CreateTime=123456799 records[1].CardNo=12001 records[1].CardName=Rohit records[1].UserID=Rohit records[1].Type=Exit records[1].Method=1 ... </pre>
--	---

12.1.8 Query Access Control Alarm Record

Table 12-4

Syntax	<code>http://<server>/cgi-bin/recordFinder.cgi?action=find&name=AccessControlAlarmRecord[&StartTime=<startTime>&EndTime=<endTime>&count=<countNo>]</code>
Method	GET
Description	Find the AccessControlAlarmRecord record. (This api is supported by access control device.)
Example	<code>http://192.168.1.250/cgi-bin/recordFinder.cgi?action=find&name=AccessControlAlarmRecord&StartTime=2014-8-25%2000:02:32&EndTime=2014-8-25%2001:02:32&count=500</code>
Success Return	totalCount =1000 found =500 records[0].ReINDo=789 records[0].CreateTime=123456789 records[0].UserID=10113 records[0].EventCode=DoorMagnetism records[0].DevAddrs=1 records[0].IndexNum=0 records[0].Time=2017-05-10 16:00:01 ...
Comment	Parameters in URL: startTime : The start time ,format : 2014-8-25%2000:01:32 endTime : The end time, format: 2014-8-25%2000:02:32 countNo : the number of records to get, The record count, default 1024 Parameters in Response : totalCount : the record count which match condition found : the record count to return EventCode : alarm event code, the range is { "DoorNotClosed", "BreakIn", "RepeatEnter", "Duress", "AlarmLocal", "ChassisIntruded" }

12.1.9 [Event] AccessControl

Table 12-5

Usage	Refer to "4.4.3 Subscribe to Snapshot" for how to subscribe event		
Description	When user trying to open the door, send this event.		
[Event Params] (key=value format)			
Name	Type	R/O	Param Description
Events	array<object>	R	Event info array
+EventBaseInfo	object	R	Base info of event
++Code	String	R	Event Code. It should be AccessControl
++Action	String	R	Event Action. It can be: "Start", "Stop", "Pulse"
++Index	int	O	The channel index relate to this event.
+RelNDo	int	R	The record id
+Name	string	O	The name of the door
+Type	string	O	The event type. It can be: " Entry ", " Exit "
+Status	int	O	Open door result, can be: 0—failed, 1—success If this param does not exist, that means success.

+Method	int	R	<p>Open door method, can be:</p> <p>0: by password 1: by access card 2: by access card and then password 3: by password and then access card 6: by fingerprint 7: by password and access card and fingerprint together 8 : by password and fingerprint together 9 : by access card and fingerprint together 10: reserved 11: by multiple access user 12: by key 13: by duress password 14: by QR code, local 15: by face recognition, local 16: reserved 17: by ID card 18: by face and ID card 19: by Bluetooth 20: by custom password 21: by UserId and password 22: by face and password 23: by fingerprint and password 24: by fingerprint and face 25: by access card and face 26: by face or password 27: by fingerprint or password 28: by fingerprint or face 29: by access card or face 30: by access card or fingerprint 31: by fingerprint and face and password 32: by access card and face and password 33: by access card and fingerprint and password 34: by access card and fingerprint and face 35: by fingerprint or face or password 36: by access card or face or password 37: by access card or fingerprint or face 38: by access card and fingerprint and face and password 39: by access card or fingerprint or face or password 40: by ID card and face, or access card or face 41: by ID card or QR code or face 42: by DTMF(SIPINFO, RFC2833, INBAND) 43: by QR code, remote 44: by face recognition, remote 45: by ID card (match with fingerprint in ID card) 46: by temporary password 47: by health code </p>
---------	-----	---	--

+CardNo	string	O	Card number if the door is opened by card
+UserID	string	R	The user id
+ErrorCode	integer	O	error code
+ObjectProperties	object	O	Dynamic structure info
+Mask	integer	O	is with mask, it can be { 0: unknow 1: without mask 2: with mask }
+ManTemperatu reInfo	object	O	people's temperature info
++CurrentTempe rature	float	O	people's current temperature
++TemperatureU nit	integer	O	temperature unit:{ 0: centigrade 1: Fahrenheit 2: Kelvin }
++IsOverTemper ature	bool	O	is over temperature

[Example]

Event	<pre>--<boundary> Content-Type: text/plain Content-Length: <length> Events[0].EventBaseInfo.Code=TrafficJam Events[0].EventBaseInfo.Action=Pulse Events[0].EventBaseInfo.Index=0 Events[0].RelNDo=123 Events[0].Name=Door1 Events[0].Type=Entry Events[0].Status=1 Events[0].Method=1 Events[0].CardNo=09DDAABB Events[0].UserID=101 ... --<boundary> Content-Type: image/jpeg Content-Length: <image size> <Jpeg image data> --<boundary></pre>
-------	--

12.1.10 [Event] CitizenPictureCompare

Table 12-6

Usage	Refer to "4.4.3 Subscribe to Snapshot" for how to subscribe event		
Description	When user trying to open the door, send this event.		
[Event Params] (key=value format)			
Name	Type	R/O	Param Description
Events	array<object>	R	Event info array
+EventBaseInfo	object	R	Base info of event
++Code	String	R	Event Code. It should be CitizenPictureCompare
++Action	String	R	Event Action. It can be: "Pulse"
++Index	int	O	The channel index relate to this event.
+UTC	uint32	O	UTC time
+CompareResult	bool	O	compare result. If the similarity is greater than or equal to the threshold, the comparison is considered successful
+Similarity	integer	O	Similarity of two pictures, Unit: percentage range[1, 100]
+Threshold	integer	O	check threshold range[1, 100]
+Citizen	String	O	citzen name
+Sex	integer	O	sex, it can be: 0 unknow 1 male 2 female 9 not stated
+Minzu	integer	O	Minzu (refer to CitizenIDCard)
+Birth	String	O	birthday, such as "1980-01-01"
+Address	String	O	address
+Number	String	O	citizen ID
+Authority	String	O	signing and issuing organization
+Start	String	O	Effective date
+End	String	O	Expiration date, and "Endless" mens Long term effectiveness
+ImageInfo	array<object>	O	Picture information, the first for the face cutout, the second for the ID card photo
++Offset	integer	O	the offset in binary data
++Length	integer	O	length of picture, unit: Byte
++Width	integer	O	Picture width, pixels
++Height	integer	O	Picture height, pixels
+ImageInfoEx	array<object>	O	extension of picture info, (The total number of images uploaded is determined by imageinfo and imageinfoex) 6 at most
++Type	integer	O	picture type 0 Local face database

			1 Shooting scene map
++Offset	integer	O	the offset in binary data
++Length	integer	O	length of picture, unit: Byte
++Width	integer	O	Picture width, pixels
++Height	integer	O	Picture height, pixels
+CardNo	String	O	IC card number (for building products)
+CellPhone	String	O	phone number (input number before comparison) (for building products)
+BuildingNumber	String	O	building number (for building products)
+BuildingUnitNumber	String	O	building entrance number (for building products)
+BuildingRoomNumber	String	O	room number (for building products)
+PersonnelRelationship	String	O	relationship (for building products)
+Method	integer	O	method (refer to OpenDoorMethod)
+EventGroupID	integer	O	event group id, used to associate different events are with same action. (Used with AccessControlevents)
+CallNumber	String	O	phone number to be called.(called after compare succeed) (for building products)
+EventType	integer	O	event type: 0: compare result 1: Face acquisition 2: visitor registration 3: Face permission distribution 4: inquirement of ID card face group
+UserID	String	O	user ID
+FaceIndex	integer	O	face index, range [0,4]
+Mask	integer	O	is with mask, it can be { 0: unknow 1: without mask 2: with mask } default 0.

[Example]

Event	--<boundary> Content-Type: text/plain Content-Length: <length> Events[0].EventBaseInfo.Code= CitizenPictureCompare Events[0].EventBaseInfo.Action=Pulse Events[0].EventBaseInfo.Index=0 Events[0].UTC=1999999999 Events[0].CompareResult=true Events[0].Similarity=80
-------	---

	<pre>Events[0].Threshold=75 ... --<boundary> Content-Type: image/jpeg Content-Length: <image size> <Jpeg image data> --<boundary></pre>
--	--

12.1.11 [Event] Door Status Event

Usage	Refer to "4.4.3 Subscribe to Snapshot" for how to subscribe event.		
Description	Door status event		
[Event Params] (key=value format)			
Name	Type	R/O	Param Description
Events	array<object>	R	Event array
+EventBaseInfo	object	R	Basic information of event
++Code	String	R	Event code ("DoorStatus")
++Action	String	R	Event action ("Pulse")
++Index	int	O	Door channel
+UTC	uint32	-	Standard UTC time (without DST deviation of time zone). Required for access control products, and optional for intercom products.
+Status	enumchar[32]	-	Door status Enumchar[32]{ "Open" "Close" "CloseAlways" "OpenAlways" "Normal" }
Example			
Event	<pre>--<boundary> Content-Type: text/plain Content-Length: <length> Events[0].EventBaseInfo.Code= DoorStatus Events[0].EventBaseInfo.Action=Pulse Events[0].EventBaseInfo.Index=0 Events[0].UTC=1999999999 Events[0].Status= CloseAlways --<boundary></pre>		

12.1.12 [Config] Access Control General Setting

- Get access control general setting

Table 12-7

URL	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=AccessControlGeneral					
Method	GET					
Description	Get access control general setting.					
[Request Params] (None)						
[Response Params] (key=value format)						
Name	Type	R/O	Param Description			
table	object	R	config table object			
+AccessControlGeneral	object	R	AccessControlGeneral config object			
++AccessProperty	string	O	Access property, can be : "unidirect", "bidirect" (This param is supported by access control device.)			
+ABLock	object	O	AB Lock setting (This param is supported by access control device.)			
+++Enable	bool	O	Enable AB Lock or not.			
+++Doors	Array<Array<int>>	O	AB Lock groups, each group has several doors, one door can be opened only when all other doors in the AB lock group are in closed state.			
++CustomPasswordEnable	bool	O	Whether to enable custom password. (This param is supported by video talk device.)			
++CommonPassword	string	O	The common password. (This param is supported by video talk device.)			
++ButtonExitEnable	bool	O	Whether to enable the open door button. (This param is supported by video talk device.)			
++CheckSensorBeforeLock	bool	O	Whether to check the sensor before lock the door. (This param is supported by video talk device.)			
++CheckSensorTime	int	O	The check sensor time, in seconds. (This param is supported by video talk device.)			
++DuressPassword	string	O	The duress password. (This param is supported by video talk device.)			
++DuressEnable	bool	O	Whether to enable duress password. (This param is supported by video talk device.)			
++UnlockRecordType	Array<string>	O	Unlock Record Type,can be {"Password", "Button", "FingerPrint", "QRCode", "Card", "Remote", "BlueTooth", "Face"}			
++SensorType	integer	O	sensor type, 0 for always open; 1 for always close.			
++CallLiftType	string	O	call lift protocol type			
++CallLiftEnable	bool	O	whether to enable call lift			
++AccessVoice	object	O	Door opening prompt tone			
+++CurrentVoiceID	integer	O	the current prompt tone ID, default value 0			
+++VoiceList	array<object>	O	voice list, max length is 16 voice ID can be: {			

			0: Validate succeed 1: door open succeed 2: unlock succeed 3: punch time clocks succeed 4: welcome 5: Welcome to come again 6: thank you 7: custom voice }
++++VoiceID	integer	O	voice id
++++VoiceName	string	O	voice info
++++FileName	string	O	file full path with name
[Example]			
Request	http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=AccessControlGeneral		
Response	table.AccessControlGeneral.AccessProperty=bidirect table.AccessControlGeneral.ABLock.Enable=true table.AccessControlGeneral.ABLock.Doors[0][0]=1 table.AccessControlGeneral.ABLock.Doors[0][1]=2 table.AccessControlGeneral.ABLock.Doors[0][2]=3 table.AccessControlGeneral.ABLock.Doors[1][0]=4 table.AccessControlGeneral.ABLock.Doors[1][1]=5 table.AccessControlGeneral.ABLock.Doors[1][2]=6 table.AccessControlGeneral.CustomPasswordEnable=true table.AccessControlGeneral.CommonPassword=123456 table.AccessControlGeneral.ButtonExitEnable=true table.AccessControlGeneral.CheckSensorBeforeLock=true table.AccessControlGeneral.CheckSensorTime=30 table.AccessControlGeneral.DuressPassword=654321 table.AccessControlGeneral.DuressEnable=true		

- Set access control general setting

Table 12-8

URL	http://<server>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set access control general setting.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
AccessControlGeneral	object	R	AccessControlGeneral config object
+AccessProperty	string	O	Access property, can be : "unidirect", "bidirect" (This param is supported by access control device.)
...<other param>	—	—	...<See above getConfig command for other params of the config object, They are all optional.>
[Response Params] (OK)			
[Example]			

Request	http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&AccessControlGeneral.AccessProperty=bidirect&AccessControlGeneral.ABLock.Enable=false&AccessControlGeneral.Cu stomPasswordEnable=true&AccessControlGeneral.CommonPassword=123456&AccessControlGeneral.ButtonExitEnable=true&AccessControlGeneral.CheckSensorBeforeLock=true
Response	OK

12.1.13 [Config] Access Control Setting

- Get access control setting

Table 12-9

URL	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=AccessControl					
Method	GET					
Description	Get access control setting.					
[Request Params] (None)						
[Response Params] (key=value format)						
Name	Type	R/O	Param Description			
table	object	R	The config table			
+AccessControl	array<object>	R	The AccessControl config array for every access control channel			
++Enable	bool	O	Whether to enable config for this channel.			
++State	string	O	Door state, can be : "Normal", "CloseAlways", "OpenAlways", "NoPersonNC", "NoPersonNO"			
++Method	int	O	Open door method, can be following value, default is 2 : 0 : only by password 1 : only by access card 2 : by password or access card 3 : by access card first then password 4 : by password first then access card 5 : different method in differenct time range 6 : only by fingerprint 7 : by password or access card or fingerprint 8 : by password and access card and fingerprint together 9 : by password and fingerprint together 10 : by access card and fingerprint together 11 : by multiple access user 12 : by face (match with picture in ID card) 13 : by face and ID card 14 : by face (match with picture in ID card) or access card or fingerprint 15 : by face and ID card, or access card or fingerprint 16 : by UserID and password 17 : only by face 18 : by face and password together			

			<p>19 : by fingerprint and password together 20 : by fingerprint and face together 21 : by access card and face together 22 : by face or password 23 : by fingerprint or password 24 : by fingerprint or face 25 : by access card or face 26 : by access card or fingerprint 27 : by fingerprint and face and password together 28 : by access card and face and password together 29 : by access card and fingerprint and password together 30 : by access card and fingerprint and face together 31 : by fingerprint or face or password 32 : by access card or face or password 33 : by access card or fingerprint or face 34 by access card and fingerprint and face and password together 35 : by access card or fingerprint or face or password 36 : by face and ID card, or access card or face 37 : by face (match with picture in ID card) or access card or face 38 : by access card and password, or fingerprint and password 39 : by ID card (match with picture in ID card) or face 40 : by ID card (match with fingerprint in ID card) 41 : by ID card (match with fingerprint and picture in ID card) 42 : by ID card or access card or fingerprint or face or password 43 : by multy user method 44 : by ID card or health code</p>
++OpenAlwaysTime	int	O	The time range that working, value is index in AccessTimeSechdule config.
++CloseAlwaysTime	int	O	The time range that always closed, value is index in AccessTimeSechdule config.
++HolidayTime	int	O	The time range that working in holidays, value is index in AccessTimeSechdule config.
++UnlockHoldInterval	int	O	The Unlock holding interval, unit is milliseconds, value should between 250 to 20000. (This param is also supported by video talk device.)
++UnlockReloadInterval	int	O	The Unlock reload interval, unit is milliseconds. (This param is also supported by video talk device.)
++AccessProtocol	string	O	The access control process, can be : "Local", "Cpplus", "Remote", "Private" (This param is also supported by video talk device.)
++BreakInAlarmEnable	bool	O	Whether to enable the breakin alarm.

++RepeatEnterAlarm	bool	O	Whether to enable the repeat enter alarm.
++DoorNotClosedAlarmEnable	bool	O	Whether to enable the alarm when door not closed.
++DuressAlarmEnable	bool	O	Whether to enable the duress alarm.
++FirstEnter	object	O	The first enter setting.
+++Enable	bool	O	Whenther to enable the first enter function,
+++Status	string	O	The status, can be : "KeepOpen" "Normal"
+++Time	int	O	The time range that enable first enter function, value is index in AccessTimeSechdule config.
++CardNoConvert	int	O	The card number convert setting : 0 – not convert, 1 – convert using NOT operation, 2 : use HIDpro convert
++MaliciousAccessControlEnable	bool	O	Whether to enable malicious access alarm.
+AutoRemoteCheck	object	O	auto open door remotely
+++Enable	bool		whether to enable the function
+++Time	uint		working period , value is index in AccessTimeSechdule config.
++Name	string	O	name
++RepeatEnterTime	integer	O	Repeat entry time, unit second, 0 means do not start
++CloseTimeout	integer	O	Closing overtime
++SensorEnable	bool	O	sensor enable
++TimeSchedule	Array<Array<object>>		Open the door in different periods it works when Method is "different method in differenct time range".
+++TimeSection	string	O	time section, format as "hh:mm:ss-hh:mm:ss"
+++Method	integer	O	Open door method, can be following value, default is 2 : 0 : only by password 1 : only by access card 2 : by password or access card 3 : by access card first then password 4 : by password first then access card 5 : different method in differenct time range 6 : only by fingerprint 7 : by password or access card or fingerprint 8 : by password and access card and fingerprint together 9 : by password and fingerprint together 10 : by access card and fingerprint together 11 : by multiple access user 17 : only by face 35 : by access card or fingerprint or face or password
++CustomPass	bool	O	whether to enable admin password

wordEnable			
++RemoteCheck	bool	O	Is platform validation required
[Example]			
Request	http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=AccessControl		
Response	table.AccessControl[0].Enable=true table.AccessControl[0].State=Normal table.AccessControl[0].Method=2 table.AccessControl[0].OpenAlwaysTime=1 table.AccessControl[0].CloseAlwaysTime=2 table.AccessControl[0].UnlockHoldInterval=500 table.AccessControl[0].AccessProtocol=Local table.AccessControl[0].BreakInAlarmEnable=true table.AccessControl[0].DuressAlarmEnable=true table.AccessControl[0].CardNoConvert=0 table.AccessControl[0].MaliciousAccessControlEnable=true ...		

- Set access control setting

Table 12-10

URL	http://<server>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set access control setting.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
AccessControl	array<object>	R	AccessControl config object array
+Enable	bool	O	Whether to enable config for this channel.
...<other param>	—	—	...<See above getConfig command for other params of the config object, They are all optional.>
[Response Params] (OK)			
[Example]			
Request	http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&AccessControl[0].Enable=true&AccessControl[0].State=Normal&AccessControl[0].Method=2&AccessControl[0].OpenAlwaysTime=1&AccessControl[0].CloseAlwaysTime=2&AccessControl[0].UnlockHoldInterval=500&AccessControl[0].AccessProtocol=Local&AccessControl[0].BreakInAlarmEnable=true&AccessControl[0].DuressAlarmEnable=true&AccessControl[0].CardNoConvert=0&AccessControl[0].MaliciousAccessControlEnable=true		
Response	OK		

12.1.14 [Config] Wiegand Setting

- Get Wiegand setting

Table 12-11

URL	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=Wiegand					
Method	GET					
Description	Get Wiegand setting.					
[Request Params] (None)						
[Response Params] (key=value format)						
Name	Type	R/O	Param Description			
table	object	R	config table object			
+Wiegand	array<object>	R	Wiegand config			
++Mode	int	R	Wiegand mode, 0 : wiegand input, 1 : wiegand output			
++PulseWidth	int	R	The pulse width, unit is microseconds			
++PulseStep	int	R	The pulse step, unit is microseconds			
++TransferMode	int	R	The transfer mode, 0 : wiegand 34bit transfer, 4 byte card number, 2 bit checksum, 1 : wiegand 66bit transfer, 8 byte card number, 2 bit checksum, 2 : wiegand 26bit transfer, 3 byte card number, 2 bit checksum,			
++OutType	int	R	The output type, 0 : output ID, 1 : output card number			
++InputType	integer	O	the input type, the value is 32bits. It won't accept any input when value is 0. Bit0: card number input Bit1: password input bit2~Bit31:reserved			
++Doors	integer	O	door index, starts from 0, and 0 means door 1			
[Example]						
Request	http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=Wiegand					
Response	table.Wiegand[0].Mode=1 table.Wiegand[0].PulseWidth=200 table.Wiegand[0].PulseStep=1000 table.Wiegand[0].TransferMode=1 table.Wiegand[0].OutType=1 table.Wiegand[1].Mode=1 table.Wiegand[0].InputType=3 table.Wiegand[0].Doors=0 ...					

- Set Wiegand setting

Table 12-12

URL	http://<server>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set Wiegand setting.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
Wiegand	array<object>	R	Wiegand config object
+Mode	int	O	Wiegand mode, 0 : wiegand input, 1 : wiegand output
...<other param>	—	—	...<See above getConfig command for other params of the

		config object, They are all optional.>
[Response Params] (OK)		
[Example]		
Request	http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&Wiegand[0].Mode=1&Wiegand[0].PulseWidth=200&Wiegand[0].PulseStep=1000&Wiegand[0].TransferMode=1&Wiegand[0].OutType=1	
Response	OK	

12.1.15 [Config] Access Time Schedule Setting

The following Apis are not supported by video talk device.

- Get access time schedule setting

Table 12-13

URL	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=AccessTimeSchedule					
Method	GET					
Description	Get access time schedule setting.					
[Request Params] (None)						
[Response Params] (key=value format)						
Name	Type	R/O	Param Description			
table	array<object>	R	config table object			
+AccessTimeSchedule	array<object>	R	AccessTimeSchedule config object array			
++Name	string	O	The schedule name, max string length is 63.			
++Enable	bool	R	Whether to enable this time schedule			
++TimeSechdule	array< array< string > >	O	The time schedule array, first array has max 8 item, refer to 7 day in a week (the first one is Sunday)and last one is holiday, each item is an array has max 6 time section, each time section is a string, format is "enable, hour:minite:second – hour:minite:second", for example : "100:00:00-12:00:00"			
[Example]						
Request	http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=AccessTimeSchedule					
Response	table.AccessTimeSchedule[0].Name=TS1 table.AccessTimeSchedule[0].Enable=true table.AccessTimeSchedule[0].TimeSchedule[0][0]=1 00:00:00-12:00:00 table.AccessTimeSchedule[0].TimeSchedule[0][1]=1 15:00:00-20:00:00 table.AccessTimeSchedule[0].TimeSchedule[1][0]=1 00:00:00-12:00:00 table.AccessTimeSchedule[0].TimeSchedule[1][1]=1 15:00:00-20:00:00					

- Set access time schedule setting

Table 12-14

URL	http://<server>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set access time schedule setting.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
AccessTimeSchedule	array<object>	R	The access time schedule config object array
+Name	string	O	The schedule name, max string length is 63.
...<other param>	—	—	...<See above getConfig command for other params of the config object, They are all optional.>
[Response Params] (OK)			
[Example]			
Request	http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&AccessTimeSchedule[0].Name=TS1&AccessTimeSchedule[0].Enable=true&AccessTimeSchedule[0].TimeSchedule[0][0]=1%2000:00:00-12:00:00&AccessTimeSchedule[0].TimeSchedule[0][1]=1%2015:00:00-20:00:00&AccessTimeSchedule[0].TimeSchedule[1][0]=1%2000:00:00-12:00:00&AccessTimeSchedule[0].TimeSchedule[1][1]=1%2015:00:00-20:00:00		
Response	OK		

12.1.16 [Config] Special Day Group Setting

The following Apis are not supported by video talk device.

- Get special day group setting

Table 12-15

URL	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=SpecialDayGroup					
Method	GET					
Description	Get special day group setting.					
[Request Params] (None)						
[Response Params] (key=value format)						
Name	Type	R/O	Param Description			
table	object	R	config table object			
+SpecialDayGroup	array<object>	R	SpecialDayGroup config object array			
++Name	string	O	The special day group name			
++Enable	bool	R	Whether to enable this special day group			
++Days	array<object>	O	The special days in group			
+++SpecialDayName	string	O	The special day name.			
+++StartTime	string	O	The special day start datetime			
+++EndTime	string	O	The special day end datetime			
[Example]						
Request	http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=SpecialDayGroup					

Response	table.SpecialDayGroup[0].Name=SpecialDayGroup1 table.SpecialDayGroup[0].Enable=true table.SpecialDayGroup[0].Days[0].SpecialDayName=NationalDay table.SpecialDayGroup[0].Days[0].StartTime=2017-10-01 00:00:00 table.SpecialDayGroup[0].Days[0].StartTime=2017-10-07 23:59:59 ...
----------	--

- Set special day group setting

Table 12-16

URL	http://<server>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set special day group setting.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
SpecialDayGroup	array<object>	R	The special day group config object array
+Name	string	O	The special day group name
...<other param>	—	—	...<See above getConfig command for other params of the config object, They are all optional.>
[Response Params] (OK)			
[Example]			
Request	http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&SpecialDayGroup[0].Name=SpecialDayGroup1&SpecialDayGroup[0].Enable=true&SpecialDayGroup[0].Days[0].SpecialDayName=NationalDay&SpecialDayGroup[0].Days[0].StartTime=2017-10-01%2000:00:00&SpecialDayGroup[0].Days[0].StartTime=2017-10-07%2023:59:59		
Response	OK		

12.1.17 [Config] Special Days Schedule Setting

The following Apis are not supported by video talk device.

- Get special days schedule setting

Table 12-17

URL	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=SpecialDaysSchedule					
Method	GET					
Description	Get special days schedule setting.					
[Request Params] (None)						
[Response Params] (key=value format)						
Name	Type	R/O	Param Description			
table	object	R	config table object			
+SpecialDaysSchedule	array<object>	R	SpecialDaysSchedule config object array			

++Name	string	O	The special day schedule name
++Enable	bool	R	Whether to enable this special day schedule
++GroupNo	int	R	The special day group number, value is the index of SpecialDayGroup setting array.
++TimeSection	array<string>	R	The time section array, the format of the item is “enable, hour:minite:second – hour:minite:second”, for example : "1 00:00:00-12:00:00".
++Doors	array<int>	R	The doors array.

[Example]

Request	http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=SpecialDaysSchedule
Response	<pre>table.SpecialDaysSchedule[0].Name=SpecialDayGroup1 table.SpecialDaysSchedule[0].Enable=true table.SpecialDaysSchedule[0].GroupNo=1 table.SpecialDaysSchedule[0].TimeSection[0]=1 00:00:00-12:00:00 table.SpecialDaysSchedule[0].TimeSection[1]=1 15:00:00-20:00:00 table.SpecialDaysSchedule[0].Doors[0]=2 table.SpecialDaysSchedule[0].Doors[1]=3 ...</pre>

- Set special days schedule setting

Table 12-18

URL	http://<server>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set special days schedule setting.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
SpecialDaysSchedule	array<object>	R	The special days schedule config object array
+Name	string	O	The special day schedule name
...<other param>	—	—	...<See above getConfig command for other params of the config object, They are all optional.>
[Response Params] (OK)			
[Example]			
Request	<pre>http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&SpecialDaysSchedule[0].Name=SpecialDayGroup1&SpecialDaysSchedule[0].Enable=true&SpecialDaysSchedule[0].GroupNo=1&SpecialDaysSchedule[0].TimeSection[0]=1%2000:00:00-12:00:00&SpecialDaysSchedule[0].TimeSection[1]=1%2015:00:00-20:00:00&SpecialDaysSchedule[0].Doors[0]=2&SpecialDaysSchedule[0].Doors[1]=3</pre>		
Response	OK		

12.1.18 [Config] MeasureTemperature Setting

The following Apis are not supported by video talk device.

- Get measure temperature setting

Table 12-19

URL	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=MeasureTemperatur e					
Method	GET					
Description	Get measure temperature setting.					
[Request Params] (None)						
[Response Params] (key=value format)						
Name	Type	R/O	Param Description			
table	object	R	config table object			
+MeasureTemperature	object	R	MeasureTemperature config object			
++Enable	bool	R	whether to enable measure temperature function			
++MaskOpt	uint	R	mask mode 0: don not detect mask 1: mask prompt 2: mask intercept			
++OnlyTemperatureMode	bool	O	whether to enable the single function, temperature monitoring			
++TemperatureDisplay	bool	R	whether to show the temperature			
++TemperatureUnit	uint	O	temperature unit:{ 0: centigrade 1: Fahrenheit }			
++Type	integer	R	Temperature monitoring method, it can be: 0: Single chip microcomputer infrared temperature monitoring 1: the thermal imaging principle of temperature monitoring 2: Temperature monitoring module of Guide 3: single point temperature monitoring at wrist			
++GuideModuleParam	object	O	param of temperature monitoring module of Guide			
+++Threshold	float	O	the temprerature threshold, unit is °C			
+++CalibrationModel	uint32	O	Calibration Model, it can be: 0: indoor mode 1: wall mounting mode 2: Gate mode 3: exclusive floor mode			
+++Correct	Float	O	temperature correct value, unit is °C			
+++DebugModelEnable	Bool	O	whether to enable debug mode of temperature monitoring, show temperature on the top of face			
+++HeatDisplayEnbale	bool	O	whether to show heat			
+++MaxDistance	uint32	O	the max distance for temperature monitoring, unit: cm			

+++ProjectDebugModel	bool	O	whether to enable project debug mode
+++RectEnable	Bool	O	whether to show the rectangle of temperature monitoring area on the video.
+++TempRandReplaceThreshold	Float	O	the threshold under which the temperature value will be replaced with valid one. When threshold value is 0, the function won't work.
+++ValidTemperatureLowerLimit	float	O	the lower limit of valid temperature value
++InfraredTemperatureParam	object	O	param of infrared temperature monitoring
+++Correct	float	O	temperature correct value, unit is °C
+++DebugModelEnable	bool	O	whether to enable debug mode of temperature monitoring, show temperature on the top of face
+++MaxDistance	Uint32	O	the max distance for temperature monitoring, unit: cm
+++OverTemperatureMaxDistance	Uint32	O	Maximum distance for reporting high temperature events, In this distance, the high temperature was measured and reported directly. Beyond this distance, the high temperature was measured, it will indicates closing to the point, and then measured again (CM)
+++RectEnable	bool	O	whether to show the rectangle of temperature monitoring area on the video.
+++RetentionTime	Uint32	O	temperature retention Time, unit: ms
+++SensorType	string	O	sensor type in temperature monitoring module "90641", "90640"
+++Threshold	Uint32	O	temperature threshold, unit : °C
+++ValidTemperatureLowerLimit	Float	O	the lower limit of valid temperature value
++ThermalImagingParam	object	O	param of thermalImaging
+++RetentionTime	Uint32	O	temperature retention Time, unit: ms
+++Threshold	Uint32	O	temperature threshold, unit : °C
++WristTemperatureParam	Object	O	param of wrist temperature monitoring
+++Correct	Float	O	temperature correct value, unit is °C
+++InvalidTemperatureDistance	Uint32	O	invalid temperature monitoring distance
+++TemperatureTimeout	Uint32	O	timeout of temperature monitoring, unit: s
+++Threshold	Float	O	temperature threshold, unit : °C
+++ValidTemperatureDistance	Uint32	O	valid temperature monitoring distance,
+++ValidTemper	float	O	the lower limit of valid temperature value

atureLowerLimit			
[Example]			
Request	http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name= MeasureTemperature		
Response	table.MeasureTemperature.Enable=true table.MeasureTemperature.GuideModuleParam.CalibrationModel=1 table.MeasureTemperature.GuideModuleParam.Correct=0 table.MeasureTemperature.GuideModuleParam.DebugModelEnable=false table.MeasureTemperature.GuideModuleParam.HeatDisplayEnbale=false table.MeasureTemperature.GuideModuleParam.MaxDistance=0 table.MeasureTemperature.GuideModuleParam.ProjectDebugModel=false table.MeasureTemperature.GuideModuleParam.RectEnable=true table.MeasureTemperature.GuideModuleParam.TempRandReplaceThreshold=0 table.MeasureTemperature.GuideModuleParam.Threshold=37.300000 table.MeasureTemperature.GuideModuleParam.ValidTemperatureLowerLimit=30 table.MeasureTemperature.InfraredTemperatureParam.Correct=0 table.MeasureTemperature.InfraredTemperatureParam.DebugModelEnable=false table.MeasureTemperature.InfraredTemperatureParam.MaxDistance=100 table.MeasureTemperature.InfraredTemperatureParam.OverTemperatureMaxDistance=100 table.MeasureTemperature.InfraredTemperatureParam.RectEnable=true table.MeasureTemperature.InfraredTemperatureParam.RetentionTime=500 table.MeasureTemperature.InfraredTemperatureParam.SensorType= table.MeasureTemperature.InfraredTemperatureParam.Threshold=37.300000 table.MeasureTemperature.InfraredTemperatureParam.ValidTemperatureLowerLimit=30 table.MeasureTemperature.MaskOpt=0 table.MeasureTemperature.OnlyTemperatureMode=false table.MeasureTemperature.TemperatureDisplay=true table.MeasureTemperature.TemperatureUnit=0 table.MeasureTemperature.ThermallImagingParam.RetentionTime=30 table.MeasureTemperature.ThermallImagingParam.Threshold=60 table.MeasureTemperature.Type=3 table.MeasureTemperature.WristTemperatureParam.Correct=0 table.MeasureTemperature.WristTemperatureParam.InvalidTemperatureDistance=24 table.MeasureTemperature.WristTemperatureParam.TemperatureTimeout=10 table.MeasureTemperature.WristTemperatureParam.Threshold=37.300000 table.MeasureTemperature.WristTemperatureParam.ValidTemperatureDistance=5 table.MeasureTemperature.WristTemperatureParam.ValidTemperatureLowerLimit=35 ...		

- Set measure temperature setting

Table 12-20

URL	http://<server>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set measure temperature setting.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description

MeasureTemperature	object	R	The measure temperature config object
+Name	string	O	The measure temperature name
...<other param>	—	—	...<See above getConfig command for other params of the config object, They are all optional.>
[Response Params] (OK)			
[Example]			
Request	http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&MeasureTemperature.Enable=true&MeasureTemperature.GuideModuleParam.Threshold=38.5&MeasureTemperature.MskOpt=2		
Response	OK		

12.1.19 [Config] CitizenPictureCompare Setting

- Get citizen picture compare setting

Table 12-21

URL	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=CitizenPictureCompare					
Method	GET					
Description	Get citizen picture compare setting					
[Request Params] (None)						
[Response Params] (key=value format)						
Name	Type	R/O	Param Description			
table	object	R	config table object			
+CitizenPictureCompare	object	R	CitizenPictureCompare config object			
++Threshold	uint8	R	threshold of citizen picture compare, range [1, 100]			
++UnlockEnable	bool	R	whether enable unlock if false, it won't unlock, when citizen picture compare succeed.			
++FuncEnable	bool	O	whether to enable citizen picture comparison			
++CitizenIDCheck	bool	O	is citizen ID check supported			
++SysMode	string	O	device work mode, it can be: { "FaceCollect" , "VisitorVerify", "CitizenCompare", "CitizenIDCheck", "Other" }			
[Example]						
Request	http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name= CitizenPictureCompare					
Response	table.CitizenPictureCompare.SysMode=CitizenCompare table.CitizenPictureCompare.Threshold=50 table.CitizenPictureCompare.UnlockEnable=false table.CitizenPictureCompare.FuncEnable=false table.CitizenPictureCompare.CitizenIDCheck=false ...					

- Set citizen picture compare setting

Table 12-22

URL	http://<server>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set citizen picture compare setting		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
CitizenPictureCompare	object	R	CitizenPictureCompare config object
...<other param>	—	—	...<See above getConfig command for other params of the config object, They are all optional.>
[Response Params] (OK)			
[Example]			
Request	http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&CitizenPictureCompare.SysMode= CitizenIDCheck&CitizenPictureCompare.Threshold=80&CitizenPictureCompare.UnlockEnable=true&CitizenPictureCompare.FuncEnable=true&CitizenPictureCompare.CitizenIDCheck=true		
Response	OK		

12.2 Access Control Manager

12.2.1 Get Access Control Capability

Table 12-23

URL	http://<server>/cgi-bin/accessControlManager.cgi?action=getCaps					
Method	GET					
Description	Get access control capabilities.					
[Request Params] (none)						
[Response Params] (key=value)						
caps	object	R	The access control capabilities.			
+AccessControlChannels	int	R	The access control channel number.			
+AccessControlAlarmRecord	bool	O	Support log access control alarm record or not. (This param is not supported by video talk device)			
+CustomPasswordEncryption	int	O	The custom password crypt type, 0 : plain text, 1 : MD5			
+SupportFingerPrint	int	O	Support fingerprint type, 0 : unknown, 1 : not support, 2 : support			
+OnlySingleDoorAuth	int	O	Support single door auth type, 0 : not support, 1 : support, (This param is not used by video talk device)			

+AsynAuth	int	O	Support async auth type, 0 : not support, 1 : support, (This param is not supported by video talk device)
+SpecialDaysSchedule	object	O	Special days schedule capabilities, (This param is not supported by video talk device)
++Support	bool	O	Support special days schedule or not. (This param is not supported by video talk device)
++MaxSpecialDaysSchedules	uint	O	Max special days schedule number. (This param is not supported by video talk device)
++MaxTimePeriodsPerDay	uint	O	Max time periods per day. (This param is not supported by video talk device)
++MaxSpecialDayGroups	uint	O	Max special day groups. (This param is not supported by video talk device)
++MaxDaysInSpecialDayGroup	uint	O	Max special days in special day group. (This param is not supported by video talk device)
[Example]			
Request	http://192.168.1.250/cgi-bin/accessControlManager.cgi?action=getCaps		
Response	caps.AccessControlChannels=3 caps.AccessControlAlarmRecord=true caps.CustomPasswordEncryption=1 caps.SupportFingerPrint=1 caps.OnlySingleDoorAuth=1 caps.AsynAuth=1 caps.SpecialDaysSchedule.Support=true caps.SpecialDaysSchedule.MaxSpecialDaysSchedules=5 caps.SpecialDaysSchedule.MaxTimePeriodsPerDay=5 caps.SpecialDaysSchedule.MaxSpecialDayGroups=5 caps.SpecialDaysSchedule.MaxDaysInSpecialDayGroup=5		

12.2.2 Add SubController

Table 12-24

Syntax	<a href="http://<server>/cgi-bin/accessControlManager.cgi?action=addDevice&devAddr=<devAddr>&name=<name>">http://<server>/cgi-bin/accessControlManager.cgi?action=addDevice&devAddr=<devAddr>&name=<name>
Method	GET
Description	Add SubController.
Example	http://192.168.1.250/cgi-bin/accessControlManager.cgi?action=addDevice&devAddr=1&name=Sub1
Success Return	FaiedCode=0
Comment	Params in URL: devAddr: SubController ID, range is [1...255] name: SubController Name Params in Response:

	FaiedCode: 0-Success; 1-Already exists; 2-Reach quantity limit; 3-Other reason
--	---

12.2.3 Modify SubController

Table 12-25

Syntax	http://<server>/cgi-bin/accessControlManager.cgi?action=modifyDevice&devAddr=< devAddr >&name=< name >
Method	GET
Description	Modify SubController.
Example	http://192.168.1.250/cgi-bin/accessControlManager.cgi?action=modifyDevice&devAddr=1&name=Sub11
Success Return	OK
Comment	Params in URL: devAddr: SubController ID, range is [1...255] name: SubController Name

12.2.4 Remove SubController

Table 12-26

Syntax	http://<server>/cgi-bin/accessControlManager.cgi?action=removeDevice&devAddrs[0]=< devAddr0 >[& devAddrs[1]=< devAddr1 >&...&devAddrs[n]=< devAddrn >]
Method	GET
Description	Remove SubController.
Example	http://192.168.1.250/cgi-bin/accessControlManager.cgi?action=removeDevice&devAddrs[0]=4&devAddrs[1]=5
Success Return	devInfo[0].DevAddr=4 devInfo[0].FaiedCode=0 devInfo[1].DevAddr=5 devInfo[1].FaiedCode=1
Comment	Params in URL: devAddr0...devAddrn: SubController ID, range is [1...255] Params in Response: DevAddr: the removed SubController ID FaiedCode: 0-Success; 1-Fail; 2-Other

12.2.5 Get Sub Controller Info

Table 12-27

Syntax	http://<server>/cgi-bin/accessControlManager.cgi?action=getSubControllerInfo&devAd drs[0]=< devAddr0 >& devAddrs[1]=< devAddr1 >&...&devAddrs[n]=< devAddrn >]
Method	GET
Description	Get SubController Info.
Example	http://192.168.1.250/cgi-bin/accessControlManager.cgi?action=getSubControllerInfo&d evAddrs[0]=4&devAddrs[1]=5
Success Return	devInfo[0].DevAddr=4 devInfo[0].Name=Sub4 devInfo[0].AccessProperty=unidirect devInfo[0].DeviceType=ACS-1202B-S devInfo[0].VerSion=V1.00428 devInfo[0].DoorToReadID[0].Door=0 devInfo[0].DoorToReadID[0].ReaderID[0]=1 devInfo[0].DoorToReadID[0].ReaderID[1]=2 devInfo[0].DoorToReadID[1].Door=1 devInfo[0].DoorToReadID[1].ReaderID[0]=3 devInfo[0].DoorToReadID[1].ReaderID[1]=4 devInfo[1].DevAddr=5 devInfo[1].Name=Sub5 devInfo[1].AccessProperty=bidirect devInfo[1].DeviceType=ACS-1202B-S devInfo[1].VerSion=V1.00428 devInfo[1].DoorToReadID[0].Door=0 devInfo[1].DoorToReadID[0].ReaderID[0]=11 devInfo[1].DoorToReadID[0].ReaderID[1]=12
Comment	<p>Params in URL:</p> <p>devAddr0...devAddrn: SubController ID, range is [1...255]</p> <p>Params in Response:</p> <p>DevAddr: SubController ID, range is [1...255]</p> <p>AccessProperty: range is [unidirect, bidirect]</p> <p>DoorToReadID: ReaderID used by the door</p>

12.2.6 Get Sub Controller States

Table 12-28

Syntax	http://<server>/cgi-bin/accessControlManager.cgi?action=getSubControllerStates&dev Addrs[0]=< devAddr0 >& devAddrs[1]=< devAddr1 >&...&devAddrs[n]=< devAddrn >]
Method	GET
Description	Get SubController States.
Example	http://192.168.1.250/cgi-bin/accessControlManager.cgi?action=getSubControllerStates &devAddrs[0]=4&devAddrs[1]=5
Success Return	devStates[0].DevAddr=4 devStates[0].State=0

	devStates[1].DevAddr=5 devStates[1].State=1
Comment	<p>Params in URL:</p> <p>devAddr0...devAddrn: SubController ID, range is [1...255]</p> <p>Params in Response:</p> <p>DevAddr: SubController ID, range is [1...255]</p> <p>State: 0-Offline; 1-Online; 2-Other</p>

12.2.7 Set RepeatEnter Route

Table 12-29

Syntax	http://<server>/cgi-bin/accessControlManager.cgi?action=setRepeatEnterRoute&index=< index >&routeInfo.PointInfo[x0].ReaderID[0]=< ReadID0 >&routeInfo.PointInfo[x0].ReaderID[1]=< ReadID1 >&routeInfo.PointInfo[x1].ReaderID[0]=< ReadID0 >&routeInfo.PointInfo[x1].ReaderID[1]=< ReadID1 >&routeInfo.TimeSections=< TimeIndex >&routeInfo.Flag=< Flag >&routeInfo.Name=< Name >
Method	GET
Description	Set RepeatEnter Route.
Example	http://192.168.1.250/cgi-bin/accessControlManager.cgi?action=setRepeatEnterRoute&index=0&routeInfo.PointInfo[0].ReaderID[0]=11&routeInfo.PointInfo[0].ReaderID[1]=22&routeInfo.PointInfo[1].ReaderID[0]=33&routeInfo.PointInfo[1].ReaderID[1]=44&routeInfo.TimeSections=1&routeInfo.Flag=1&routeInfo.Name=123
Success Return	OK
Comment	<p>Params in URL:</p> <p>index: route index</p> <p>x0,x1: PointInfo index</p> <p>TimeIndex: subscript of AccessTimeSchedule</p> <p>Flag: 0-Disable this route; 1-Enable this route</p> <p>Name: route name</p>

12.2.8 Get RepeatEnter Route

Table 12-30

Syntax	http://<server>/cgi-bin/accessControlManager.cgi?action=getRepeatEnterRoute&index=< index >
Method	GET
Description	Get RepeatEnter Route.
Example	http://192.168.1.250/cgi-bin/accessControlManager.cgi?action=getRepeatEnterRoute&index=0

Success Return	routeInfo.PointInfo[0].ReaderID[0]=11 routeInfo.PointInfo[0].ReaderID[1]=22 routeInfo.PointInfo[1].ReaderID[0]=33 routeInfo.PointInfo[1].ReaderID[1]=44 routeInfo.TimeSections=1 routeInfo.Flag=1 routeInfo.Name=123
Comment	<p>Params in URL:</p> <p>index: route index</p> <p>Params in Response:</p> <p>PointInfo: is an array, stands for points in route</p> <p>TimeSections: subscript of AccessTimeSchedule</p> <p>Flag: 0-Disable this route; 1-Enable this route</p> <p>Name: route name</p>

12.2.9 Set ABLOCK Route

Table 12-31

Syntax	http://<server>/cgi-bin/accessControlManager.cgi?action=setABLockRoute&index=< index >&routeInfo.PointInfo[x0].Doors[0]=< DoorID0 >&routeInfo.PointInfo[x0].Doors[1]=< DoorID1 >[&routeInfo.PointInfo[x1].Doors[0]=< DoorID0 >&routeInfo.PointInfo[x1].Doors[1]=< DoorID1 >]&routeInfo.TimeSections=< TimeIndex >&routeInfo.Flag=< Flag >&routeInfo.Name=< Name >
Method	GET
Description	Set ABLOCK Route.
Example	http://192.168.1.250/cgi-bin/accessControlManager.cgi?action=setABLockRoute&index=0&routeInfo.PointInfo[0].Doors[0]=11&routeInfo.PointInfo[0].Doors[1]=22&routeInfo.PointInfo[1].Doors[0]=33&routeInfo.PointInfo[1].Doors[1]=44&routeInfo.TimeSections=1&routeInfo.Flag=1&routeInfo.Name=123
Success Return	OK
Comment	<p>Params in URL:</p> <p>index: route index</p> <p>x0,x1: PointInfo index</p> <p>TimeIndex: subscript of AccessTimeSchedule</p> <p>Flag: 0-Disable this route; 1-Enable this route</p> <p>Name: route name</p>

12.2.10 Get ABLock Route

Table 12-32

Syntax	<code>http://<server>/cgi-bin/accessControlManager.cgi?action=getABLockRoute&index=<<i>index</i>></code>
Method	GET
Description	Get ABLock Route.
Example	<code>http://192.168.1.250/cgi-bin/accessControlManager.cgi?action=getABLockRoute&index=0</code>
Success Return	<code>routeInfo.PointInfo[0].Doors[0]=11</code> <code>routeInfo.PointInfo[0].Doors [1]=22</code> <code>routeInfo.PointInfo[1].Doors [0]=33</code> <code>routeInfo.PointInfo[1].Doors [1]=44</code> <code>routeInfo.TimeSections=1</code> <code>routeInfo.Flag=1</code> <code>routeInfo.Name=123</code>
Comment	Params in URL: <i>index</i>: route index Params in Response: PointInfo: is an array, stands for points in route TimeSections: subscript of AccessTimeSchedule Flag: 0-Disable this route; 1-Enable this route Name: route name

12.2.11 Get Log Status

Table 12-33

Syntax	<code>http://<server>/cgi-bin/accessControlManager.cgi?action=getLogStatus&devAddr=<<i>devAddr</i>></code>
Method	GET
Description	Get Log Status.
Example	<code>http://192.168.1.250/cgi-bin/accessControlManager.cgi?action=getLogStatus&devAddr=1</code>
Success Return	OK
Comment	Params in URL: <i>devAddr</i>: SubController ID, range is [1...255]

12.2.12 Sync Offline Log

Table 12-34

Syntax	http://<server>/cgi-bin/accessControlManager.cgi?action=syncOfflineLog&devAddr=< d evAddr >
Method	GET
Description	Sync Offline Log.
Example	http://192.168.1.250/cgi-bin/accessControlManager.cgi?action=syncOfflineLog&devAd dr=1
Success Return	OK
Comment	Params in URL: devAddr: SubController ID, range is [1...255]

12.2.13 Sync SubController Time

Table 12-35

Syntax	http://<server>/cgi-bin/accessControlManager.cgi?action=syncSubControllerTime&dev Addrs[0]=< devAddr0 >[& devAddrs[1]=< devAddr1 >&...&devAddrs[n]=< devAddrn >]
Method	GET
Description	Sync SubController Time.
Example	http://192.168.1.250/cgi-bin/accessControlManager.cgi?action=syncSubControllerTime &devAddrs[0]=4&devAddrs[1]=5
Success Return	errorDevAddrs[0]=4
Comment	Params in URL: devAddr0...devAddrn: SubController ID, range is [1...255] Params in Response: errorDevAddrs: is an array, stands for DevAddr which failed in sync time

12.3 Access User Account (V1)

12.3.1 Add Access User Face

Table 12-36

URL	http://<server>/cgi-bin/FaceInfoManager.cgi?action=add		
Method	POST		
Description	Add an access user face.		
[Request Params] (JSON format in body)			
Name	Type	R/O	Param Description
UserID	string	R	The user's id.
Info	object	R	The user's info.
+UserName	string	O	The user's name.
+RoomNo	array<string>	O	The user's room num list, max array size is 32, max string size is 12. (This param is not supported by access control.)
+FaceData	array<string>	O	The user's face feature, encode with base64, max array size is 20, and max string size is 2k. Note: There must be at least one between FaceData and PhotoData.
+PhotoData	array<string>	O	The user's face photo, encode with base64, max array size is 5, and max string size is 200k. Note: There must be at least one between FaceData and PhotoData.
[Response Params] (JSON format in body)			
[Example]			
Request	<pre>POST http://<server>/cgi-bin/FaceInfoManager.cgi?action=add Content-Type: application/json Content-Length: <length> { "UserID": "102", "Info": { "UserName": "Rohit", "RoomNo": ["301", "303", ...], "FaceData": ["xxxx", "xxxx", ...], "PhotoData": ["yyyy", "yyyy", ...] } }</pre>		
Response	{}		

12.3.2 Modify Access User Face

Table 12-37

URL	http://<server>/cgi-bin/FaceInfoManager.cgi?action=update
-----	---

Method	POST		
Description	Update an access user face.		
[Request Params] (JSON format in body)			
Name	Type	R/O	Param Description
UserID	string	R	The user's id.
Info	object	R	The user's info.
+UserName	string	O	The user's name.
+RoomNo	array<string>	O	The user's room num list, max array size is 32, max string size is 12. (This param is not supported by access control.)
+FaceData	array<string>	O	The user's face feature, encode with base64, max array size is 20, and max string size is 2k. Note: There must be at least one between FaceData and PhotoData.
+PhotoData	array<string>	O	The user's face photo, encode with base64, max array size is 5, max string size is 200k. Note: There must be at least one between FaceData and PhotoData.
[Response Params] (JSON format in body)			
[Example]			
Request	<pre>POST http://<server>/cgi-bin/FaceInfoManager.cgi?action=update Content-Type: application/json Content-Length: <length> { "UserID": "102", "Info": { "UserName": "Rohit", "RoomNo": ["301", "303", ...], "FaceData": ["xxxx", "xxxx", ...], "PhotoData": ["yyyy", "yyyy", ...] } }</pre>		
Response	{}		

12.3.3 Delete Access User Face

Table 12-38

URL	http://<server>/cgi-bin/FaceInfoManager.cgi?action=remove		
Method	GET		
Description	Delete an access user face.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
UserID	string	R	The user's id.
[Response Params] (OK)			
[Example]			
Request	GET http://<server>/cgi-bin/FaceInfoManager.cgi?action=remove&UserID=102		

Response	OK
----------	----

12.3.4 Find Access User Face

- Start to find access user face

Table 12-39

URL	http://<server>/cgi-bin/FaceInfoManager.cgi?action=startFind		
Method	GET		
Description	Start to find access user face.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
Condition	object	O	Search condition.
+UserID	string	O	The user's id.
[Response Params] (JSON format)			
Name	Type	R/O	Param Description
Token	int	O	The token of this search, use this token to get result and stop search.
Total	int	O	Result num, return 0 if not found.
[Example]			
Request	GET http://<server>/cgi-bin/FaceInfoManager.cgi?action=startFind		
Response	{ "Token": 1234, "Total": 20 }		

- Get the find result

Table 12-40

URL	http://<server>/cgi-bin/FaceInfoManager.cgi?action=doFind		
Method	GET		
Description	Get the find result.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
Token	int	R	The token of this search, use this token to get result and stop search.
Offset	int	R	Offset in the result record set, range is [0, Total – 1].
Count	int	R	Count of result to get.
[Response Params] (JSON format)			
Name	Type	R/O	Param Description
Info	array<object>	R	The face information of user.
+UserID	string	R	The user's id.
+MD5	array<string>	O	The user's face photo's MD5 hash string, max array size is 5, max string size is 33.
[Example]			

Request	GET http://<server>/cgi-bin/FaceInfoManager.cgi?action=doFind&Token=1234&Offset=0&Count=20
Response	{ "Info": [{ "UserID": "102", "MD5": ["xxxx", "xxxx", ...] }, { ... }, ...] }

- Stop the find session

Table 12-41

URL	http://<server>/cgi-bin/FaceInfoManager.cgi?action=stopFind		
Method	GET		
Description	Stop the find session.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
Token	int	R	Token for this search, use this token to get result and stop search.
[Response Params] (OK)			
[Example]			
Request	GET http://<server>/cgi-bin/FaceInfoManager.cgi?action=stopFind&Token=1234		
Response	OK		

12.3.5 Add Access User Card and Fingerprint

- Add Access user card only

Table 12-42

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=insert&name=AccessControlCard		
Method	GET		
Description	Insert access user card info.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access user card and fingerprint record name, should be " AccessControlCard ".
CardName	string	R	Access user card name, max string length is 32.
CardNo	string	R	Access user card number.
UserID	string	R	The user's id.

CardStatus	int	R	Card status, 0 means normal, and abnormal status can be sum of following: <ul style="list-style-type: none"> • 1 << 0: report the loss of the card • 1 << 1: the card is withdrawn • 1 << 2: the card is frozen • 1 << 3: the card is arrearage • 1 << 4: the card exceeds the time limit • 1 << 5: the card is arrearage but still can open the door, and there will be voice prompts.
CardType	int	O	The card type: 0 : Normal Card, 1: VIP Card, 2: Visitor Card, 3: Patrol Card, 4: Blocklist Card, 5: Stress Card, 0xff: Mother Card
Password	string	O	The Access card's password. (This param is not supported by video talk device)
Doors	array<int>	O	The index of the doors that this card can open. (This param is not supported by video talk device)
TimeSections	array<int>	O	The index of the time sections of each door that this card can open. (This param is not supported by video talk device)
VTOPosition	string	O	VTO position number. (This param is not supported by access control device)
ValidDateStart	string	O	The start of valid date, format is "yyyyMMdd hhmmss".
ValidDateEnd	string	O	The end of valid date, format is "yyyyMMdd hhmmss".
IsValid	bool	O	Is the card still valid. (This param is not supported by video talk device)

[Response Params] (key=value format)

Name	Type	R/ O	Param Description
ReINDo	int	R	The record id.

[Example]

Request	GET http://192.168.1.250/cgi-bin/recordUpdater.cgi?action=insert&name=AccessControlCard&CardName=Rohit&CardNo=12345&UserID=102&CardStatus=0&CardType=0&Password=123456&Doors[0]=1&Doors[1]=3&Doors[2]=5&VTOPosition=01018001&ValidDateStart=20151022%20093811&ValidDateEnd=20151222%20093811
Response	ReINDo=12345

- Add access user card with fingerprint

Add access user card with fingerprint. If add only access user card, please use :

<http://<server>/cgi-bin/recordUpdater.cgi?action=insert&name=AccessControlCard>

Table 12-43

URL	<a href="http://<server>/cgi-bin/recordUpdater.cgi?action=insertEx&name=AccessControlCard">http://<server>/cgi-bin/recordUpdater.cgi?action=insertEx&name=AccessControlCard		
Method	POST		
Description	Insert access user card and fingerprint. (This api is not supported by video talk device)		
[Request Params] (key=value format in URL)	Type	R/O	Param Description
Name	Type	R/O	Param Description

name	string	R	Access user card and fingerprint record name, should be "AccessControlCard".
CardName	string	O	Access user card name, max string length is 32.
CardNo	string	O	Access user card number.
UserID	string	R	The user's id.
CardStatus	int	O	Card status, 0 means normal, and abnormal status can be sum of following: <ul style="list-style-type: none"> • 1 << 0: report the loss of the card • 1 << 1: the card is withdrawn • 1 << 2: the card is frozen • 1 << 3: the card is arrearage • 1 << 4: the card exceeds the time limit • 1 << 5: the card is arrearage but still can open the door, and there will be voice prompts.
CardType	int	O	The card type: 0: Normal Card, 1: VIP Card, 2: Visitor Card, 3: Patrol Card, 4: Blocklist Card, 5: Stress Card, 0xff: Mother Card
Password	string	O	The Access card's password.
Doors	array<int>	O	The index of the doors that this card can open.
TimeSections	array<int>	O	The index of the time sections of each door that this card can open.
VTOPosition	string	O	VTO position number. (This param is not supported by access control device)
ValidDateStart	string	O	The start of valid date, format is "yyyyMMdd hhmmss".
ValidDateEnd	string	O	The end of valid date, format is "yyyyMMdd hhmmss".
IsValid	bool	O	Is the card still valid.
FingerprintPacket	object	O	The fingerprint packet info.
+Length	int	O	The length of each fingerprint packet.
+Count	int	O	The count of fingerprint packet.

[Response Params] (key=value format)

Name	Type	R/ O	Param Description
RelINDo	int	R	The record id.

[Example]

Request	<p>POST</p> <p>http://192.168.1.250/cgi-bin/recordUpdater.cgi?action=insertEx&name=AccessControlCard&CardName=Rohit&CardNo=12345&UserID=102&CardStatus=0&Password=123456&Doors[0]=1&Doors[1]=3&Doors[2]=5&VTOPosition=01018001&ValidDateStart=20151022%20093811&ValidDateEnd=20151222%20093811&FingerprintPacket.Length=500&FingerprintPacket.Count=3</p> <p>Content-Type: application/octet-stream</p> <p>Content-Length: <length></p> <p><fingerprint packet binary data></p>
Response	RelINDo=12345

12.3.6 Modify Access User Card and Fingerprint

- Update Access user card only

Table 12-44

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=update&name=AccessControlCard		
Method	GET		
Description	Update access user card info. Note: You should provide at least one optional param to update.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access user card and fingerprint record name, should be " AccessControlCard ".
reINDo	int	R	The record id.
CardName	string	O	Access user card name, max string length is 31.
CardNo	string	O	Access user card number. Primary key, modification is not allowed in principle
UserID	string	O	The user's id. Primary key, modification is not allowed in principle
...<other param>	—	—	...<See above insert command for other params of the record, They are all optional.>
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.250/cgi-bin/recordUpdater.cgi?action=update&name=AccessControlCard&reINDo=12345&CardName=Rohit&CardNo=12345&UserID=102&CardStatus=0&CardType=0&Password=123456&Doors[0]=1&Doors[1]=3&Doors[2]=5&ValidDateStart=20151022%20093811&ValidDateEnd=20151222%20093811		
Response	OK		

- Update access user card and fingerprint

Table 12-45

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=updateEx&name=AccessControlCard					
Method	POST					
Description	Update access user card info and fingerprint. (This api is not supported by video talk device)					
Note: You should provide at least one optional param to update.						
[Request Params] (key=value format in URL)						
Name	Type	R/O	Param Description			
name	string	R	Access user card and fingerprint record name, should be " AccessControlCard ".			
reINDo	int	R	The record id.			
CardName	string	O	Access user card name, max string length is 32.			

CardNo	string	O	Access user card number. Primary key, modification is not allowed in principle
UserID	string	O	The user's id. Primary key, modification is not allowed in principle
...<other param>			...<See above insert command for other params of the record. They are all optional.>
FingerprintPacket	object	O	The fingerprint packet info.
+Length	int	O	The length of each fingerprint packet.
+Count	int	O	The count of fingerprint packet.
[Response Params] (OK)			
[Example]			
Request	POST http://192.168.1.250/cgi-bin/recordUpdater.cgi?action=updateEx&name=AccessControlCard&reINDo=12345&CardName=Rohit&CardNo=12345&UserID=102&CardStatus=0&CardType=0&Password=123456&Doors[0]=1&Doors[1]=3&Doors[2]=5&ValidDateStart=20151022%20093811&ValidDateEnd=20151222%20093811&FingerprintPacket.Length=500&FingerprintPacket.Count=3 Content-Type: application/octet-stream Content-Length: <length> <fingerprint packet binary data>		
Response	OK		

12.3.7 Delete Access User Card and Fingerprint

- Delete Access user card and fingerprint record by reINDo

Table 12-46

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=remove&name=AccessControlCard		
Method	GET		
Description	Remove the access user card and fingerprint record. (The video talk device can not delete fingerprint data.)		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access user card and fingerprint record name, should be "AccessControlCard".
reINDo	int	R	The record id.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.250/cgi-bin/recordUpdater.cgi?action=remove&name=AccessControlCard&reINDo=12345		
Response	OK		

- Delete all Access user card and fingerprint records

Table 12-47

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=clear&name=AccessControlCard		
Method	GET		
Description	Remove all the access user card and fingerprint records. (The video talk device can not delete fingerprint data.)		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access user card and fingerprint record name, should be " AccessControlCard ".
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.250/cgi-bin/recordUpdater.cgi?action=clear&name=AccessControlCard		
Response	OK		

12.3.8 Find Access User Card and Fingerprint

- Find Access user card by condition

Table 12-48

URL	http://<server>/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCard		
Method	GET		
Description	Find Access user card record by condition.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Record Name, should be " AccessControlCard ".
count	int	O	Max result to return, default is 1024.
StartTime	string	O	The start of the record's CreateTime.
EndTime	string	O	The End of the record's CreateTime.
condition	object	O	Search condition.
+CardNo	string	O	Access user card number.
+UserID	string	O	The user's id.
+IsValid	bool	O	The access card valid or not.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
totalCount	int	O	The total record num that find.
found	int	O	The record num that returned.
records	array<object>	R	The records that returned.
+ReIDNo	int	R	The record id.
+CreateTime	int	O	The create time of record.
+CardName	string	R	The access user card name, max string length is 32.
+CardNo	string	R	The access user card number.
+UserID	string	R	The user's id.

+CardStatus	int	R	Card status, 0 means normal, and abnormal status can be sum of following: <ul style="list-style-type: none">• 1 << 0: report the loss of the card• 1 << 1: the card is withdrawn• 1 << 2: the card is frozen• 1 << 3: the card is arrearage• 1 << 4: the card exceeds the time limit• 1 << 5: the card is arrearage but still can open the door, and there will be voice prompts.
+CardType	int	O	The card type: 0: Normal Card, 1: VIP Card, 2: Visitor Card, 3: Patrol Card, 4: Blocklist Card, 5: Stress Card, 0xff: Mother Card
+Password	string	O	The Access card's password. (This param is not supported by video talk device)
+Doors	array<int>	O	The index of the doors that this card can open. (This param is not supported by video talk device)
+TimeSections	array<int>	O	The index of the time sections of each door that this card can open. (This param is not supported by video talk device)
+VTOPosition	string	O	VTO position number. (This param is not supported by access control device)
+ValidDateStart	string	O	The start of valid date, format is "yyyyMMdd hhmmss".
+ValidDateEnd	string	O	The end of valid date, format is "yyyyMMdd hhmmss".
+IsValid	bool	O	Is the card still valid. (This param is not supported by video talk device)
+Handicap	bool	O	is the card of extend time
+CitizenIDNo	string	O	citizen card id
+RepeatEnterRouteTimeout	integer	O	timeout of repeater enter route, unit: s
+UserType	integer	O	user type, it can be { 1: normal user 2: user in Blocklist }
+FirstEnter	bool	O	is with first enter authority
+DynamicCheckCode	string	O	dynamic check code
+UseTime	integer	O	count of use
[Example]			
Request	GET http://192.168.1.250/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCard&condition.UserID=103&StartTime=123456700&EndTime=123456800&count=100		

	totalCount=1000 found=100 records[0].reINDo=1234 5 records[0].CreateTime=123456789 records[0].CardName=Rohit records[0].CardNo=300 records[0].UserID=103 records[0].CardStatus=0 records[0].CardType=0 records[0].Doors[0]=1 records[0].Doors[1]=3 records[0].Doors[2]=5 records[0].VTOPosition=01018001 records[0].ValidStart=20151022 093811 records[0].ValidEnd=20151222 093811 records[0].isValid=true ... records[1].reINDo=1357 9 records[1].CreateTime=123456799 records[1].StaticsticsTime=123456799 records[1].CardName=Rohit records[1].CardNo=302 records[1].UserID=103 records[1].CardStatus=0 records[1].CardType=0 records[1].Doors[0]=2 records[1].Doors[1]=4 records[1].Doors[2]=6 records[1].VTOPosition=01018002 records[1].ValidStart=20151022 093811 records[1].ValidEnd=20151222 093811 records[1].isValid=true ...
Response	

- Find Access user card by reINDo

Table 12-49

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=get&name=AccessControlCard		
Method	GET		
Description	Find Access user card record by reINDo.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Record Name, should be " AccessControlCard ".
reINDo	int	R	The record id.
[Response Params] (key=value format)			

Name	Type	R/O	Param Description
record	object	R	The record that returned.
+RelINDo	int	R	The record id.
+CreateTime	int	O	The create time of record.
+CardName	string	R	The access user card name, max string length is 32.
+CardNo	string	R	The access user card number.
+UserID	string	R	The user's id.
...<other param>	—	—	...<See above find command for other params of the record.>

[Example]

Request	GET http://192.168.1.250/cgi-bin/recordUpdater.cgi?action=get&name=AccessControlCard&relINDo=4
Response	record.RelINDo=4 record.CreateTime=123456789 record.CardName=Rohit record.CardNo=300 record.UserID=103 record.CardStatus=0 record.CardType=0 record.Doors[0]=1 record.Doors[1]=3 record.Doors[2]=5 record.VTOPosition=01018001 record.ValidStart=20151022 093811 record.ValidEnd=20151222 093811 record.IsValid=true ...

- Find Access user card and fingerprint by relINDo

Table 12-50

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=getEx&name=AccessControlCard		
Method	GET		
Description	Find Access user card and fingerprint record by relINDo. (This api is not supported by video talk device)		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Record Name, should be " AccessControlCard ".
relINDo	int	R	The record id.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
record	object	R	The record that returned.
+RelINDo	int	R	The record id.
+CreateTime	int	O	The create time of record.
+CardName	string	R	The access user card name, max string length is 32.
+CardNo	string	R	The access user card number.
+UserID	string	R	The user's id.

...<other param>	—	—	...<See above find command for other params of the record.>
+FingerprintPacket	object	O	The fingerprint packet info.
++Length	int	O	The length of each fingerprint packet.
++Count	int	O	The count of fingerprint packet.
[Example]			
Request	GET http://192.168.1.250/cgi-bin/recordUpdater.cgi?action=getEx&name=AccessControlCard&reINDo=4		
Response	HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<boundary> Content-Length: <length> --<boundary> Content-Type: text/plain Content-Length: <length> record.ReINDo=4 record.CreateTime=123456789 record.CardName=Rohit record.CardNo=300 record.UserID=103 record.CardStatus=0 record.CardType=0 record.Doors[0]=1 record.Doors[1]=3 record.Doors[2]=5 record.VTOPosition=01018001 record.ValidStart=20151022 093811 record.ValidEnd=20151222 093811 record.IsValid=true ... record.FingerprintPacket.Length=500 record.FingerprintPacket.Count=3 --<boundary> Content-Type: application/octet-stream Content-Length: <length> <fingerprint packet binary data> --<boundary>--		

12.3.9 Get the Total Number of Records of Access User Card and Fingerprint

Table 12-51

URL	http://<server>/cgi-bin/recordFinder.cgi?action=getQuerySize&name=AccessControlCard		
Method	GET		
Description	Get the total number of records of the access user card record.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access user card and fingerprint record name, should be "AccessControlCard".
[Response Params] (key=value)			
count	int	R	The total number of records.
[Example]			
Request	GET http://192.168.1.250/cgi-bin/recordFinder.cgi?action=getQuerySize&name=AccessControlCard		
Response	count=150		

12.4 Access User Account (V2)

12.4.1 Add Access User

- Insert or update multiple new users. Update when the current user exists. This only applies to the AccessUser entity here.

Table 12-52

URL	http://<server>/cgi-bin/AccessUser.cgi?action=insertMulti		
Method	POST		
Description	Insert person information.		
[Request Params] (JSON format in body)			
Name	Type	R/O	Param Description
UserList	array<object>	R	User list, with up to 10 entries
+UserID	string	R	User ID
+UserName	string	O	User name
+UserType	uint16	O	0: General user, by default; 1: Blocklist user (report the blocklist event ACBlocklist); 2: Guest user; 3: Patrol user; 4: VIP user; 5: Extend time user
+UseTime	uint16	O	Limit of passing times for guest users
+IsFirstEnter	bool	O	First user authority or not. false: No; true: Yes
+FirstEnterDoors	int16[]	O	-1 indicates all channels.
+UserStatus	uint16	O	0: Normal; 1: Frozen
+Authority	uint8	O	User authority (attendance machine field). 1: Admin; 2: General user
+CitizenIDNo	string	O	ID card number
+Password	string	O	The password when unlocking by card + password.

			The password when unlocking by User ID + password
+Doors	int16[]	O	Door authority. The index in the controller is used with TimeSections, and the value corresponds to the subscript of the AC configuration.
+TimeSections	uint16[]	O	The door authority corresponds to the period index. For example, door 3 corresponds to period 2. Each element corresponds to the door in Doors.
+SpecialDaysSchedule	uint32[64]	O	Holiday plan identification. The value is the subscript number configured by SpecialDaysSchedule (defined in the configuration).
+ValidFrom	string	O	"yyyy-MM-dd HH:mm:ss", start of validity period. Note: The original "ValidDateStart" is deprecated.
+ValidTo	string	O	"yyyy-MM-dd HH:mm:ss", end of validity period. Note: The original "ValidDateEnd" is deprecated.

[Response Params] (OK)

[Example]

Request	<pre>{ UserList:[{"UserID": "100013", "UserName": "", "UserType": 0, "UseTime": 1, "IsFirstEnter": true, "FirstEnterDoors": [0, 1], "UserStatus": 0, "Authority": 1, "CitizenIDNo": "123456789012345678", "Password": "xxxxxxxxxx", "Doors": [1,3,5,7], "TimeSections": [1,2,3,4], "SpecialDaysSchedule": [1,2], "ValidFrom": "2018-01-02 00:00:00", "ValidTo": "2018-01-02 01:00:00", } ,...,{}] } }</pre>
Response	OK

12.4.2 Modify Access User

Update information of multiple users.

Table 12-53

URL	http://<server>/cgi-bin/AccessUser.cgi?action=updateMulti
Method	POST

Description		Update person information.	
[Request Params] (JSON format in body)			
Name	Type	R/O	Param Description
UserList	array<object>	R	User list, with up to 10 entries
+ UserID	string	R	User ID
+UserName	string	O	User name
+UserType	uint16	O	0: General user, by default; 1: Blocklist user (report the blocklist event ACBlocklist); 2: Guest user; 3: Patrol user; 4: VIP user; 5: Extend time user
+UseTime	uint16	O	Limit of passing times for guest users
+IsFirstEnter	bool	O	First user authority or not. false: No; true: Yes
+FirstEnterDoors	int16[]	O	-1 indicates all channels.
+UserStatus	uint16	O	0: Normal; 1: Frozen
+Authority	uint8	O	User authority (attendance machine field). 1: Admin; 2: General user
+CitizenIDNo	string	O	ID card number
+Password	string	O	The password when unlocking by card + password. The password when unlocking by User ID + password
+Doors	int16[]	O	Door authority. The index in the controller is used with TimeSections, and the value corresponds to the subscript of the AC configuration.
+TimeSections	uint16[]	O	The door authority corresponds to the period index. For example, door 3 corresponds to period 2. Each element corresponds to the door in Doors.
+SpecialDaysSchedule	uint32[64]	O	Holiday plan identification. The value is the subscript number configured by SpecialDaysSchedule (defined in the configuration).
+ValidFrom	string	O	"yyyy-MM-dd HH:mm:ss", start of validity period. Note: The original "ValidDateStart" is deprecated.
+ValidTo	string	O	"yyyy-MM-dd HH:mm:ss", end of validity period. Note: The original "ValidDateEnd" is deprecated.
[Response Params] (OK)			
[Example]			
Request	<pre>{ UserList:[{"UserID": "100013", "UserName": "", "UserType": 0, "UseTime": 1, "IsFirstEnter": true, "FirstEnterDoors": [0, 1], "UserStatus": 0, "Authority": 1, "CitizenIDNo": "123456789012345678", "Password": "xxxxxxxxxx",</pre>		

	<pre> "Doors": [1,3,5,7], "TimeSections": [1,2,3,4], "SpecialDaysSchedule": [1,2], "ValidFrom": "2018-01-02 00:00:00", "ValidTo": "2018-01-02 01:00:00", } ,...,{}] } </pre>
Response	OK

12.4.3 Delete All Access User

Delete all user information.

Table 12-54

URL	http://<server>/cgi-bin/AccessUser.cgi?action=removeAll		
Method	GET		
Description	Clear information of all persons.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/AccessUser.cgi?action=removeAll		
Response	OK		

12.4.4 Delete Multiple Access Users

Delete information of multiple users.

Table 12-55

URL	http://<server>/cgi-bin/AccessUser.cgi?action=removeMulti		
Method	GET		
Description	Delete user data.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
UserIDList	array<string>	R	User ID list, with up to 10 entries
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/AccessUser.cgi?action=removeMulti&UserIDList[0]=102&UserIDList[1]=102		
Response	OK		

12.4.5 Find Multiple Access Users

Search for information of multiple users.

Table 12-56

URL	http://<server>/cgi-bin/AccessUser.cgi?action=list		
Method	GET		
Description	Search for user data.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
UserIDList	array<string>	R	User ID list, with up to 10 entries
[Response Params]			
Users	array<object>	R	The records that returned
+ UserID	string	R	User ID
+UserName	string	O	User name
+UserType	uint16	O	0: General user, by default; 1: Blocklist user (report the blocklist event ACBlocklist); 2: Guest user; 3: Patrol user; 4: VIP user; 5: Extend time user
+UseTime	uint16	O	Limit of passing times for guest users
+IsFirstEnter	bool	O	First user authority or not. false: No; true: Yes
+FirstEnterDoors	int16[]	O	-1 indicates all channels
+UserStatus	uint16	O	0: Normal; 1: Frozen
+Authority	uint8	O	User authority (attendance machine field). 1: Admin; 2: General user
+CitizenIDNo	string	O	ID card number
+Password	string	O	The password when unlocking by card + password. The password when unlocking by UserID + password
+Doors	int16[]	O	Door authority. The index in the controller is used with TimeSections, and the value corresponds to the subscript of the AC configuration.
+TimeSections	uint16[]	O	The door authority corresponds to the period index. For example, door 3 corresponds to period 2. Each element corresponds to the door in Doors.
+SpecialDaysSchedule	uint32[64]	O	Holiday plan identification. The value is the subscript number configured by SpecialDaysSchedule (defined in the configuration).
+ValidFrom	string	O	"yyyy-MM-dd HH:mm:ss", start of validity period. Note: The original "ValidDateStart" is deprecated.
+ValidTo	string	O	"yyyy-MM-dd HH:mm:ss", end of validity period. Note: The original "ValidDateEnd" is deprecated.
[Example]			
Request	http://192.168.1.250/cgi-bin/AccessUser.cgi?action=list&UserIDList[0]=102&UserIDList[1]=103		
Respons	Users[0].UserID=100013		

e	Users[0].UserName=Name Users[0].UserType=1 Users[0].UseTime=1 Users[0].IsFirstEnter=Rohit Users[0].FirstEnterDoors=0 Users[0].UserStatus=12345678 Users[0].Authority=1 Users[0].CitizenIDNo=1 Users[0].Password=Rohit Users[0].Doors=0 Users[0].TimeSections=12345678 Users[0].SpecialDaysSchedule=1 Users[0].ValidFrom=1 records[0].ValidTo=Rohit ...
---	--

12.4.6 Start Find Access User Related Information

Start searching for user related information.

Table 12-57

URL	http://<server>/cgi-bin/AccessUser.cgi?action=startFind		
Method	GET		
Description	Start searching for user related information.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
Condition	object	R	Search conditions. Users can perform conditional search according to the user information field.
+UserID	string	O	User ID
+UserName	string	O	User name
+UserType	uint16	O	0: General user, by default; 1: Blocklist user (report the blocklist event ACBlocklist); 2: Guest user; 3: Patrol user; 4: VIP user; 5: Extend time user
+UseTime	uint16	O	Limit of passing times for guest users
+IsFirstEnter	bool	O	First user authority or not. false: No; true: Yes
+FirstEnterDoors	int16[]	O	-1 indicates all channels.
+UserStatus	uint16	O	0: Normal; 1: Frozen
+Authority	uint8	O	User authority (attendance machine field). 1: Admin; 2: General user
+CitizenIDNo	string	O	ID card number
+Password	string	O	The password when unlocking by card + password. The password when unlocking by UserID + password
+Doors	int16[]	O	Door authority. The index in the controller is used with TimeSections, and the value corresponds to the subscript of the AC configuration.
+TimeSections	uint16[]	O	The door authority corresponds to the period index. For

			example, door 3 corresponds to period 2. Each element corresponds to the door in Doors.
+SpecialDaysSchedule	uint32[64]	O	Holiday plan identification. The value is the subscript number configured by SpecialDaysSchedule (defined in the configuration).
+ValidFrom	string	O	"yyyy-MM-dd HH:mm:ss", start of validity period. Note: The original "ValidDateStart" is deprecated.
+ValidTo	string	O	"yyyy-MM-dd HH:mm:ss", end of validity period. Note: The original "ValidDateEnd" is deprecated.
[Response Params]			
Token	uint32	R	Search token.
Total	uint32	R	Total number of entries found this time
Caps	uint32	R	Search capability: Maximum number of records that can be returned each time.
[Example]			
Request	http://192.168.1.250/cgi-bin/AccessUser.cgi?action=startFind&Condition.UserID=1		
Response	<pre>{ "Token": 1234, "Total": 20, "Caps ": 20 }</pre>		

12.4.7 Get Find Result of Access User Related Information

Get user related information.

Table 12-58

URL	http://<server>/cgi-bin/AccessUser.cgi?action=doFind		
Method	GET		
Description	Get user related information.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
Token	int	R	Search token
Offset	int	R	Offset
Count	int	R	Number of entries obtained this time
[Response Params]			
info	array<object>	R	Person information
[Example]			
Request	http://192.168.1.250/cgi-bin/AccessUser.cgi?action=doFind&Token=1234&Offset=0&Count=20		
Response	<pre>{ "Info": [{ "UserID": "102", ... }, { ... }, ...] }</pre>		

12.4.8 Stop Find Access User Related Information

Get user related information.

Table 12-59

URL	http://<server>/cgi-bin/AccessUser.cgi?action=stopFind		
Method	GET		
Description	Get user related information.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
Token	int	R	Search token.
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/AccessUser.cgi?action=stopFind&Token=1234		
Response	OK		

12.4.9 Add Multiple Access Cards

Send information of multiple card numbers.

Table 12-60

URL	http://<server>/cgi-bin/AccessCard.cgi?action=insertMulti		
Method	POST		
Description	Insert card number information.		
[Request Params] (JSON format in body)			
Name	Type	R/O	Param Description
CardList	array<object>	R	User list, with up to 10 entries
+CardNo	string	R	Card No.
+UserID	string	R	User ID
+CardType	uint16	O	Card type Enumint{ 0: Ordinary card 1: VIP card 2: Guest card 3: Patrol card 4: Blocklist card 5: Duress card}
+CardName	string	O	Card name
+CardStatus	uint32	O	Card status. Different card status results in different person status. 0: Normal 1<<0: Reported for loss 1<<1: Canceled 1<<2: Frozen

			1<<3: Arrearage 1<<4: Overdue
[Response Params] (OK)			
[Example]			
Request	{ CardList[{ "UserID" : "100013" "CardNo" : "" "CardType" : 0 "CardName" : "201 - Rohit" "CardStatus" : 0 } ,...,{ }] }		
Response	OK		

12.4.10 Modify Multiple Access Cards

Update information of multiple card numbers.

Table 12-61

URL	http://<server>/cgi-bin/AccessCard.cgi?action=updateMulti		
Method	POST		
Description	Update card number information.		
[Request Params] (JSON format in body)			
Name	Type	R/O	Param Description
CardList	array<object>	R	User list, with up to 10 entries
+CardNo	string	R	Card No.
+UserID	string	R	User ID
+CardType	uint16	O	Card type Enumint{ 0: Ordinary card 1: VIP card 2: Guest card 3: Patrol card 4: Blocklist card 5: Duress card
+CardName	string	O	Card name
+CardStatus	uint32	O	Card status. Different card status results in different person status. 0: Normal 1<<0: Reported for loss 1<<1: Canceled 1<<2: Frozen 1<<3: Arrearage 1<<4: Overdue
[Response Params] (OK)			

[Example]	
Request	<pre>{ CardList:["UserID" : "100013" "CardNo" : "" "CardType" : 0 "CardName" : "201 - Rohitg" "CardStatus" : 0 } ,...,{} }</pre>
Response	OK

12.4.11 Delete All Access Cards

Remove all card number information.

Table 12-62

URL	http://<server>/cgi-bin/AccessCard.cgi?action=removeAll		
Method	GET		
Description	Clear all card information.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/AccessCard.cgi?action=removeAll		
Response	OK		

12.4.12 Delete Multiple Access Cards

Remove information of multiple card numbers.

Table 12-63

URL	http://<server>/cgi-bin/AccessCard.cgi?action=removeMulti		
Method	GET		
Description	Delete card number data.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
CardNoList	array<string>	R	User card number list, with up to 10 entries
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/AccessCard.cgi?action=removeMulti&CardNoList[0]=12345678&CardNoList[1]=12345687		
Response	OK		

12.4.13 Find Multiple Access Cards

Search for information of multiple card numbers.

Table 12-64

URL	http://<server>/cgi-bin/AccessCard.cgi?action=list		
Method	GET		
Description	Search for card number data.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
CardNoList	array<string>	R	User ID list, with up to 10 entries
[Response Params]			
records	array<object>	R	The records that returned.
+CardNo	string	R	Card No.
+UserID	string	R	User ID
+CardType	uint16	O	Card type Enumint{ 0: Ordinary card 1: VIP card 2: Guest card 3: Patrol card 4: Blocklist card 5: Duress card
+CardName	string	O	Card name
+CardStatus	uint32	O	Card status. Different card status results in different person status. 0: Normal 1<<0: Reported for loss 1<<1: Canceled 1<<2: Frozen 1<<3: Arrearage 1<<4: Overdue
[Example]			
Request	http://192.168.1.250/cgi-bin/AccessCard.cgi?action=list&CardNoList[0]=102&CardNoList[1]=102		
Response	Cards[0].CardNo=12345678 Cards[0].UserID=1 Cards[0].CardType=1 Cards[0].CardName=Rohit Cards[0].CardStatus=0 ... Cards[1].CardNo=12345679 Cards[1].UserID=2 Cards[1].CardType=1 Cards[1].CardName=LiSi Cards[1].CardStatus=0		

	...
--	-----

12.4.14 Start Find Access User Card Related Information

Start searching for card number related information.

Table 12-65

URL	http://<server>/cgi-bin/ AccessCard.cgi?action=startFind		
Method	GET		
Description	Start searching for card number related information.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
Condition	object	R	Search conditions. Users can perform conditional search according to the user information field.
+UserID	string	O	User ID
+CardNo	string	O	User card number
+CardType	uint16	O	Card type Enumint{ 0: Ordinary card 1: VIP card 2: Guest card 3: Patrol card 4: Blocklist card 5: Duress card
+CardName	string	O	Card name
+CardStatus	uint32	O	Card status Different card status results in different person status. 0: Normal 1<<0: Reported for loss 1<<1: Canceled 1<<2: Frozen 1<<3: Arrearage 1<<4: Overdue
[Response Params]			
Token	uint32	R	Search token.
Total	uint32	R	Total number of entries found this time
Caps	uint32	R	Search capability: Maximum number of records that can be returned each time.
[Example]			
Request	http://192.168.1.250/cgi-bin/AccessCard.cgi?action=startFind&Condition.UserID=1		
Response	{ "Token": 1234, "Total": 20, "Caps ": 20 }		

12.4.15 Get Find Result of Access User Card Related Information

Get card number related information.

Table 12-66

URL	http://<server>/cgi-bin/AccessCard.cgi?action=doFind		
Method	GET		
Description	Get card number related information.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
Token	int	R	Search token.
Offset	int	R	Offset
Count	int	R	Number of entries obtained this time
[Response Params]			
info	array<object>	R	Person information
[Example]			
Request	http://192.168.1.250/cgi-bin/AccessCard.cgi?action=doFind&Token=1234&Offset=0&Count=20		
Response	{ "Info": [{ "UserID": "102", ... }, { ... }, ...] }		

12.4.16 Stop Find Access User Card Related Information

Stop searching for card number related information.

Table 12-67

URL	http://<server>/cgi-bin/AccessCard.cgi?action=stopFind		
Method	GET		
Description	Stop searching for card number related information.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
Token	int	R	Search token.
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/AccessCard.cgi?action=stopFind&Token=1234		
Response	OK		

12.4.17 Add Multiple Access User Fingerprint

Insert multiple fingerprint information.

Table 12-68

URL	http://<server>/cgi-bin/AccessFingerprint.cgi?action=insertMulti		
Method	POST		
Description	Insert fingerprint information.		
[Request Params] (JSON format in body)			
Name	Type	R/O	Param Description
vecPackets	array<string>	R	Fingerprint feature data
AccessFingerprints	array<object>	R	Fingerprint list, with up to 1 entry
+UserID	string	R	User ID
+FingerprintPacket	object	R	Send fingerprint information list.
++Length	uint32	R	Length of a single fingerprint package
++Count	uint32	R	Number of fingerprint packages
++DataURL	array<string>	O	Either cloud storage URL of fingerprint data or Length. If the Length exists and is not 0, this field is invalid when it is subject to Length. Only HTTP URL is available now.
++DuressIndex	Uint8	R	Duress fingerprint number, with a value range of [1, count]. This field is invalid if the value is illegal. That is, there is no duress fingerprint. For example, 0 indicates no duress fingerprint.
[Response Params] (OK)			
[Example]			
Request	<pre>{ "vecPackets": ["xxxx", "xxxx", ...], "AccessFingerprints": [{ "UserID": "102", "FingerprintPacket" : { "Length" : 810, "Count" : 3, "DuressIndex" : 2 } }] }</pre>		
Response	OK		

12.4.18 Modify Access User Fingerprint

Update multiple fingerprint information.

Table 12-69

URL	http://<server>/cgi-bin/AccessFingerprint.cgi?action=updateMulti	
Method	POST	

Description	Update fingerprint information.		
[Request Params] (JSON format in body)			
Name	Type	R/O	Param Description
vecPackets	array<string>	R	Binary fingerprint data (Binary Data).
AccessFingerprints	array<object>	R	Fingerprint list, with up to 1 entry
+UserID	string	R	User ID
+FingerprintPacket	object	R	Send fingerprint information list.
++Length	uint32	R	Length of a single fingerprint package
++Count	uint32	R	Number of fingerprint packages
++DataURL	array<string>	O	Either cloud storage URL of fingerprint data or Length. If the Length exists and is not 0, this field is invalid when it is subject to Length. Only HTTP URL is available now.
++DuressIndex	Uint8	R	Duress fingerprint number, with a value range of [1, count]. This field is invalid if the value is illegal. That is, there is no duress fingerprint. For example, 0 indicates no duress fingerprint.
[Response Params] (OK)			
[Example]			
Request	<pre>{ "vecPackets": ["xxxx", "xxxx", ...], "AccessFingerprints": [{ "UserID": "102", "FingerprintPacket" : { "Length" : 810, "Count" : 3, "DuressIndex" : 2 } }] }</pre>		
Response	OK		

12.4.19 Delete All Access User Fingerprint

Delete all fingerprint information.

Table 12-70

URL	http://<server>/cgi-bin/AccessFingerprint.cgi?action=removeAll		
Method	GET		
Description	Clear all fingerprint information.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
[Response Params] (OK)			

[Example]	
Request	http://192.168.1.250/cgi-bin/AccessFingerprint.cgi?action=removeAll
Response	OK

12.4.20 Delete Multiple Access User Fingerprint

Delete fingerprint information.

Table 12-71

URL	http://<server>/cgi-bin/AccessFingerprint.cgi?action=removeMulti		
Method	GET		
Description	Delete fingerprint data.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
UserIDList	array<string>	R	User card number list, with up to 1 entry
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/AccessFingerprint.cgi?action=removeMulti&UserIDList[0]=101&UserIDList[1]=102		
Response	OK		

12.4.21 Find Access User Fingerprint

Search for user fingerprint information.

Table 12-72

URL	http://<server>/cgi-bin/AccessFingerprint.cgi?action=get		
Method	GET		
Description	Search for user fingerprint information.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
UserID	string	R	User ID list, with up to 1 entry
[Response Params]			
FingerprintData	BinaryData	R	Binary fingerprint data (Binary Data).
FingerprintPacket	object	R	Description of fingerprint data
+Length	uint32	R	Length of a single fingerprint package
+Count	uint32	R	Number of fingerprint packages
+DataURL	array<string>	O	Either cloud storage URL of fingerprint data or Length. If the Length exists and is not 0, this field is invalid when it is subject to Length. Only HTTP URL is available now.
+DuressIndex	Uint8	R	Duress fingerprint number, with a value range of [1, count]. This field is invalid if the value is illegal. That is, there is no duress fingerprint. For example, 0 indicates no duress fingerprint.

[Example]	
Request	http://192.168.1.250/cgi-bin/AccessFingerprint.cgi?action=get&UserID=1
Response	FingerprintPacket.Length=810 FingerprintPacket.Count=3 FingerprintPacket.DuressIndex=1 FingerprintData=xxx

12.4.22 Add Multiple Access User Face

Insert multiple face information.

Table 12-73

URL	http://<server>/cgi-bin/AccessFace.cgi?action=insertMulti		
Method	POST		
Description	Insert face information.		
[Request Params] (JSON format in body)			
Name	Type	R/O	Param Description
FaceList	array<object>	R	Face list, with up to 10 entries
+UserID	string	R	User ID
+FaceData	array<string>	O	Base64 of red light face templates, 8192 Bytes/per face, maximum 20.
+PhotoData	array<string>	O	Base64 of white light face images, 200kx5.
+PhotoURL	array<string>	O	Either cloud storage URL of white light face images or PhotoData. If PhotoData exists, this field is invalid when it is subject to PhotoData. Only HTTP URL is available now.
[Response Params] (OK)			
[Example]			
Request	{ "FaceList": [{ "UserID": "102", "FaceData": ["xxxx", "xxxx", ...], "PhotoData": ["yyyy", "yyyy", ...], "PhotoURL": ["yyyy", "yyyy", ...], },] }		
Response	OK		

12.4.23 Update Multiple Access User Face

Update multiple face information.

Table 12-74

URL	http://<server>/cgi-bin/AccessFace.cgi?action=updateMulti		
Method	POST		
Description	Update face information.		
[Request Params] (JSON format in body)			
Name	Type	R/O	Param Description
FaceList	array<object>	R	Face list, with up to 10 entries
+UserID	string	R	User ID
+FaceData	array<string>	O	Base64 of red light face templates, 8192 Bytes/per face, maximum 20.
+PhotoData	array<string>	O	Base64 of white light face images, 200kx5.
+PhotoURL	array<string>	O	Either cloud storage URL of white light face images or PhotoData. If PhotoData exists, this field is invalid when it is subject to PhotoData. Only HTTP URL is available now.
[Response Params] (OK)			
[Example]			
Request	<pre>{ "FaceList": [{ "UserID": "102", "FaceData": ["xxxx", "xxxx", ...], "PhotoData": ["yyyy", "yyyy", ...], "PhotoURL": ["yyyy", "yyyy", ...], }, ...] }</pre>		
Response	OK		

12.4.24 Delete All Access User Face

Delete all face information.

Table 12-75

URL	http://<server>/cgi-bin/AccessFace.cgi?action=removeAll		
Method	GET		
Description	Clear all face information.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/AccessFace.cgi?action=removeAll		
Response	OK		

12.4.25 Delete Multiple Access User Face

Delete multiple face information.

Table 12-76

URL	http://<server>/cgi-bin/AccessFace.cgi?action=removeMulti		
Method	GET		
Description	Delete face data.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
UserIDList	array<string>	R	User card number list, with up to 10 entries
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/AccessFace.cgi?action=removeMulti& UserIDList[0]=101&UserIDList[1]=102		
Response	OK		

12.4.26 Find Access User Face

Search for access user face information.

Table 12-77

URL	http://<server>/cgi-bin/AccessFace.cgi?action=list		
Method	GET		
Description	Search for user face data.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
UserIDList	array<string>	R	User ID list, with up to 10 entries
[Response Params]			
FaceDataList	array<object>	R	The records that returned.
+UserID	string	R	User ID
+FaceData	array<string>	O	Base64 of red light face templates, 8192 Bytes/per face, maximum 20.
+PhotoData	array<string>	O	Base64 of white light face images, 200kx5.
+PhotoURL	array<string>	O	Either cloud storage URL of white light face images or PhotoData. If PhotoData exists, this field is invalid when it is subject to PhotoData. Only HTTP URL is available now.
[Example]			
Request	http://192.168.1.250/cgi-bin/AccessFace.cgi?action=list&UserIDList[0]=1&UserIDList[1]=2		
Response	FaceDataList[0].UserID=1 FaceDataList[0].PhotoData[0]="xxxx", FaceDataList[0].PhotoData[1]="xxxx", , FaceDataList[0].FaceData[0]="xxxx" FaceDataList[0].FaceData[1]="xxxx",		

	<pre> ... FaceDataList[1].UserID=2 FaceDataList[1].PhotoData[0]="xxxx", FaceDataList[1].PhotoData[1]="" , FaceDataList[1].FaceData[0]="xxxx" FaceDataList[1].FaceData[1]="" , ... </pre>
--	--

12.4.27 Start Find Access User Face Related Information

Start searching for face related information.

Table 12-78

URL	http://<server>/cgi-bin/AccessFace.cgi?action=startFind		
Method	GET		
Description	Start searching for face related information.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
Condition	object	R	Search conditions. Users can perform conditional search according to the user information field.
+UserID	string	O	User ID
[Response Params]			
Token	uint32	R	Search token.
Total	uint32	R	Total number of entries found this time
Caps	uint32	R	Search capability: Maximum number of records that can be returned each time.
[Example]			
Request	http://192.168.1.250/cgi-bin/AccessFace.cgi?action=startFind&Condition.UserID=1		
Response	<pre> { "Token": 1234, "Total": 20, "Caps": 20 }</pre>		

12.4.28 Get Find Result of Access User Face Related Information

Get face related information.

Table 12-79

URL	http://<server>/cgi-bin/AccessFace.cgi?action=doFind		
Method	GET		
Description	Get face related information.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
Token	int	R	Search token.
Offset	int	R	Offset
Count	int	R	Number of entries obtained this time

[Response Params]			
info	array<object>	R	Person information
[Example]			
Request	http://192.168.1.250/cgi-bin/AccessFace.cgi?action=doFind&Token=1234&Offset=0&Count=20		
Response	{ "Info": [{ "UserID": "102", ... }, { ... }, ...] }		

12.4.29 Stop Find Access User Face Related Information

Stop searching for face related information.

Table 12-80

URL	http://<server>/cgi-bin/AccessFace.cgi?action=stopFind		
Method	GET		
Description	Stop searching for face related information.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
Token	int	R	Search token.
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/AccessFace.cgi?action=stopFind&Token=1234		
Response	OK		

12.4.30 Access Control Protocol Capability Query

Second-Generation Access Control Protocol Capability Query

Request URL	http://<server>/cgi-bin/api/AC/getCaps			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
WantMethods	bool	No	Whether to request to return the Methods list, please obtain according to actual needs.	true
WantCaps	bool	No	Whether to request to return to the capability item list, please obtain according to actual needs.	true
Request Example				
{ "WantMethods": True }				

Response Params				
Name	Type	R/O	Description	Example
Caps	object	Yes	Capability set	
+AC	object	No	AC capability set	
++Channels	uint32	No	Number of supported access control channels. The former is "AccessControlChannels".	4
++HasAlarmRecord	bool	No	Supports recording access control alarm logs. The former is "AccessControlAlarmRecord".	False
++EncryptionMethod	uint8	No	The storage method of passwords in the ACCustomPassword record set. The former is "CustomPasswordEncryption". 0: Plaintext, the default value is 0. 1: MD5.	0
++HasFingerprintAuth	uint8	No	Whether fingerprint authentication is supported. The former is "SupportFingerprint". 0: Unknown, compatible with previous one (default); 1: Not supported; 2: Supported.	0
++HasCardAuth	bool	No	Whether card authentication is supported. False: Not supported; true: Supported.	False
++HasFaceAuth	bool	No	Whether ** identification authentication is supported. False: Not supported; true: Supported.	False
++OnlySingleDoorAuth	uint8	No	Whether only single-door authorization (card issuing) multi-door controller is supported. 0: Not supported; 1: Supported	0
++IsAsynAuth	uint8	No	Whether asynchronous return authentication is supported. 0: Not supported; 1: Supported	0
++IsUserIsolate	uint8	No	Whether it is a person-card separation scheme. In the person-card separation scheme, one person can have several cards. 0: No; 1: Yes	0
++MaxInsertRate	uint16	No	The general maximum number of data can be sent at a time. Default conservative set value similar to BIOS.	10

++ScheduleCaps	object	No	The schedule capability of the device, corresponding to the former SpecialDaysSchedules.	
+++Support	bool	No	Whether the new schedule is supported. False: Not supported; true: Supported.	False
+++MaxSchedules	uint16	No	The maximum number of holiday plan capabilities supported by one access controller.	6
+++MaxTimePeriodsPerDay	uint8	No	The maximum number of time periods that can be defined in a day.	6
+++MaxSpecialDayGroups	uint16	No	The maximum number of holiday plan groups supported by the access controller.	6
+++MaxDaysInSpecialDayGroup	uint16	No	The maximum number of holidays supported by a holiday group.	16
++UnlockModes	uint16[128]	No	The combination of unlocking modes supported by the device. The element value corresponds to the former "Opening Method", namely the Method value in the AccessControl configuration.	[1, 2, ...]
++SupportBacKendAI	bool	No	Whether the AI identification mode is supported. True: Supported; false: Not supported.	True
++SupportFastImport	enumint	No	Supports fast import function Enumint { 0: Not supported 1: Supported } Not supported if it does not exist.	0
++SupportFastCheck	enumint	No	Supports quick review function (only compare userID) Enumint { 0: Not supported 1: Supported } Not supported if it does not exist.	0
++SupportRapidCheck	enumint	No	Supports rapid review function Enumint { 0: Not supported 1: Supported } Not supported if it does not exist.	0

++IncrementalImport	enumint	No	Supports incremental delivery Enumint { 0: Not supported 1: Supported } Not supported if it does not exist.	0
++FingerCompareMode	uint8	No	HasFingerprintAuth supports fingerprint authentication function, this field is valid. 0: Unknown, meaningless 1: The device only supports front-end fingerprint comparison. 2: The device only supports back-end fingerprint comparison. 3: Indicates that the device supports both front-end and back-end fingerprint comparison.	2
++SupportHelmet	enumint	No	Supports safety helmet function. (Taken from IsSupportHelmet) Enumint{ 0: Not supported 1: Supported } Not supported if it does not exist.	0
++UserNameMaxLen	uint32	No	Supports limiting name length on the device.	32
++SupportASGManager	bool	No	Supports turnstile business components (the turnstile was previously hung on the access controller as a sub-device, and the related configuration of the turnstile was written in the access control. Now divide the turnstile from the access control.) true: supported. False: Not supported.	True
++SnapPicPath	char[512]	No	The local storage directory of the door-opening snapshots; the file name is stored in the door-opening record database, and the absolute path of the snapshot can be obtained by appending this directory.	"/mnt/data/userpic/"
++FacelimagePath	char[512]	No	** The local storage directory of base pictures.	"/mnt/data/Facelimage/"
++HasIrisAuth	bool	No	Whether ** identification authentication is supported.	False
++IrisImagePath	char[512]	No	Added ** opening authentication type, and added ** base storage	"/mnt/data/Image/"

			path.	
+AccessUser	object	No	AccessUser capability set.	
++MaxInsertRate	uint16	No	Maximum number of inserts per time.	10
++MaxUsers	uint32	No	Maximum number of users that can be recorded and processed.	600
++MaxFingerprintsPerUser	uint8	No	Maximum number of fingerprints that can be recorded per person.	5
++MaxCardsPerUser	uint8	No	Maximum number of card that can be recorded per person.	5
++MaxFacesPerUser	uint8	No	Maximum number of ** photo that can be recorded per person.	1
+AccessCard	object	No	AccessCard capability set.	
++MaxInsertRate	uint16	No	Maximum number of inserts per time.	10
++MaxCards	uint32	No	Maximum storage number of cards.	600
+AccessFingerprint	object	No	-	-
++MaxInsertRate	uint16	No	Maximum number of inserts per time.	10
++MaxFingerprints	uint16	No	Maximum bytes number of single fingerprint data.	810
++MaxFingerprints	uint32	No	Fingerprint storage limit.	600
++AlgorithmVendor	uint32	No	Fingerprint algorithm manufacturer; 0: Unknown; 1: Cpplus; 2: Brmicro.	0
++AlgorithmVersion	uint32	No	Fingerprint algorithm version number; each 8 bit represents a version from high to low according to Major/Minor, for example, 1.5.2 represents as 0x0001050.	
+AccessFace	object	No	AccessFace capability set.	
++MaxInsertRate	uint16	No	Maximum number of inserts per time.	10
++MaxFaces	uint32	No	** Storage Limit.	600
++RecognitionType	uint8	No	** Recognition Type.	1
++RecognitionAlgorithmVendor	uint16	No	** Identification algorithm provider. 0: Unknown; 1: Cpplus; 2: SenseTime; 3: Yitu; 4: Hanvon; 5: Huoyan.	0
++RecognitionVersion	uint32	No	** Identification algorithm (model) version number, if the version number has multiple digits, each 8 bit represents a version from high to low according to Major/Minor, for example, 1.5.2 represents as	

			0x00010502.	
++MinPhotoSize	uint16	No	Minimum size of white light photo, KB.	20
++MaxPhotoSize	uint16	No	Maximum size of white light photo, KB.	20
++MaxGetPhotoRate	uint16	No	The maximum amount of acquisitions per time by the white light list method. Unit (number/per time).	20
++IsSupportGetPhoto	bool	No	Whether the list interface is supported to obtain white light photos.	True
++IsSupportOnlyIssueFaceEigen	bool	No	Whether only sending the characteristic value is supported.	True-
++MultiFaceDetect	object	No	-	
+++Support	bool	No	Whether multi-detection identification is supported.	True
+++MaxNums	uint32	No	The maximum number of detections supported at a time.	3
+AccessIris	object	No	AccessIris capability set.	
++MaxInsertRate	uint16	No	Maximum number of inserts per time.	10
++MinIrisPhotoSize	uint16	No	Minimum size of the image, unit KB.	1
++MaxIrisPhotoSize	uint16	No	Maximum size of the image, unit KB.	50
++MaxIrisGroup	uint16	No	Maximum number of groups supported by each user**.	1
++RecognitionAlgorithmVendor	uint16	No	** Identification algorithm provide identification, 0: Unknown; 1: Cplus.	0
++RecognitionVersion	uint32	No	** The algorithm (model) version number, if the version number has multiple digits, every 8 bit represents a version from high to low according to Major/Minor, for example, 1.5.2 represents as 0x00010502.	
++MaxIrisesCount	uint32	No	** Storage Limit.	10000
Response Example				

12.5 Admin Password

12.5.1 Add Access Control Admin Password

Table 12-81

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=insert&name=AccessControlCustomPassword		
Method	GET		
Description	Insert access control admin password.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access control admin password record name, should be " AccessControlCustomPassword ".
UserID	string	R	The user's id.
OpenDoorPassword	string	R	Ths user's open door password.
AlarmPassword	string	O	Ths user's alarm password. (This param is not supported by video talk device)
Doors	array<int>	O	The index of the doors that admin password can open. (This param is not supported by video talk device)
TimeSections	array<int>	O	The index of the time sections of each door that this card can open. (This param is not supported by video talk device)
VTOPosition	string	O	VTO position number. (This param is not supported by access control device)
ValidDateStart	string	O	The start of valid date, format is " yyyyMMdd hhmmss ".
ValidDateEnd	string	O	The end of valid date, format is " yyyyMMdd hhmmss ".
ValidCounts	int	O	The password's valid counts.
OriginSmartGateWay	string	O	The origin smart gateway address. (Video talk device customize.)
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
RelINDo	int	R	The record id.
[Example]			
Request	http://192.168.1.250/cgi-bin/recordUpdater.cgi?action=insert&name=AccessControlCustomPassword&UserID=102&OpenDoorPassword=123456&Doors[0]=1&Doors[1]=3&Doors[2]=5&VTOPosition=01018001&ValidDateStart=20151022%20093811&ValidDateEnd=20151222%20093811&ValidCounts=30		
Response	RelINDo=12345		

12.5.2 Modify Access Control Admin Password

Table 12-82

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=update&name=AccessControlCustomPassword		
Method	GET		
Description	Update access control admin password. Note: You should provide at least one optional param to update.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access control admin password record name, should be " AccessControlCustomPassword ".
reINDo	int	R	The record id.
UserID	string	R	The user's id.
OpenDoorPassword	string	R	Ths user's open door password.
Doors	array<int>	O	The index of the doors that admin password can open.
...<other param>	—	—	...<See above insert command for other params of the record, They are all optional.>
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/recordUpdater.cgi?action=update&name=AccessControlCustomPassword&reINDo=12345&UserID=102&OpenDoorPassword=123456&Doors[0]=1&Doors[1]=3&Doors[2]=5&ValidDateStart=20151022%20093811&ValidDateEnd=20151222%20093811		
Response	OK		

12.5.3 Delete Access Control Admin Password

- Delete access control admin password record by reINDo

Table 12-83

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=remove&name=AccessControlCustomPassword		
Method	GET		
Description	Remove the access control admin password record by reINDo.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access control admin password record name, should be " AccessControlCustomPassword ".
reINDo	int	R	The record id.
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/recordUpdater.cgi?action=remove&name=AccessControlCustomPassword&reINDo=12345		
Response	OK		

- Delete all the access control admin password records

Table 12-84

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=clear&name=AccessControlCustomPassword		
Method	GET		
Description	Remove all the access control admin password records		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access control admin password record name, should be " AccessControlCustomPassword ".
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/recordUpdater.cgi?action=clear&name=AccessControlCustomPassword		
Response	OK		

12.5.4 Find Access Control Admin Password

- Find Access control admin password by condition

Table 12-85

URL	http://<server>/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCustomPassword		
Method	GET		
Description	Find Access control admin password by condition.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Record Name, should be " AccessControlCustomPassword ".
count	int	O	Max result to return, default is 1024.
StartTime	string	O	The start of the record's CreateTime.
EndTime	string	O	The End of the record's CreateTime.
condition	object	O	Search condition.
+UserID	string	O	The user's id.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
totalCount	int	O	The total record num that find.
found	int	O	The record num that returned.
records	array<object>	R	The records that returned.
+RelID	int	R	The record id.
+CreateTime	int	O	The create time of record.
+UserID	string	R	The user's id.
+OpenDoorPassword	string	R	Ths user's open door password.
+AlarmPassword	string	O	Ths user's alarm password. (This param is not supported by video talk device)

+Doors	array<int>	O	The index of the doors that admin password can open. (This param is not supported by video talk device)
+TimeSections	array<int>	O	The index of the time sections of each door that this card can open. (This param is not supported by video talk device)
VTOPosition	string	O	VTO position number. (This param is not supported by video talk device)
+ValidDateStart	string	O	The start of valid date, format is "yyyyMMdd hhmmss".
+ValidDateEnd	string	O	The end of valid date, format is "yyyyMMdd hhmmss".
+ValidCounts	int	O	The password's valid counts.
+OriginSmartGateway	string	O	The origin smart gateway address. (Video talk device customize.)
[Example]			
Request	<pre>http://192.168.1.250/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCustomPassword&condition.UserID=103&StartTime=123456700&EndTime=123456800&count=100</pre>		
Response	<pre>totalCount=1000 found=100 records[0].ReINDo=1234 5 records[0].CreateTime=123456789 records[0].UserID=103 records[0].OpenDoorPassword=123456 records[0].Doors[0]=1 records[0].Doors[1]=3 records[0].Doors[2]=5 records[0].VTOPosition=01018001 records[0].ValidStart=20151022 093811 records[0].ValidEnd=20151222 093811 ... records[1].ReINDo=1357 9 records[1].CreateTime=123456799 records[1].UserID=103 records[0].OpenDoorPassword=123456 records[1].Doors[0]=2 records[1].Doors[1]=4 records[1].Doors[2]=6 records[1].VTOPosition=01018002 records[1].ValidStart=20151022 093811 records[1].ValidEnd=20151222 093811 ...</pre>		

- Find Access control admin password by reINDo

Table 12-86

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=get&name=AccessControlCustomPassword
-----	---

Method	GET		
Description	Find Access control admin password by reINDo.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Record Name, should be " AccessControlCustomPassword ".
reINDo	int	R	The record id.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
record	object	R	The record that returned.
+ReINDo	int	R	The record id.
+CreateTime	int	O	The create time of record.
+UserID	string	R	The user's id.
...<other param>	—	—	...<See above find command for other params of the record.>
[Example]			
Request	http://192.168.1.250/cgi-bin/recordUpdater.cgi?action=get&name=AccessControlCustomP assword&reINDo=3		
Response	record.ReINDo=3 record.CreateTime=123456789 record.UserID=103 record.OpenDoorPassword=123456 record.Doors[0]=1 record.Doors[1]=3 record.Doors[2]=5 record.VTOPosition=01018001 record.ValidStart=20151022 093811 record.ValidEnd=20151222 093811 ...		

12.5.5 Get the Total Number of Records of Access Control Admin Password

Table 12-87

URL	http://<server>/cgi-bin/recordFinder.cgi?action=getQuerySize&name=AccessControlCust omPassword		
Method	GET		
Description	Get the total number of records of the access control admin password.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access control admin password record name, should be " AccessControlCustomPassword ".
[Response Params] (key=value)			
count	int	R	The total number of records.
[Example]			

Request	http://192.168.1.250/cgi-bin/recordFinder.cgi?action=getQuerySize&name=AccessControlCustomPassword
Response	count=150

13.Intelligent Building API

13.1 Video Talk

The following Apis are supported by VTT / VTA products.

13.1.1 Subscribe Video Talk Status

Subscribe the video talk status. When client disconnect, it will unsubscribe.

Request URL	http://<server>/cgi-bin/VideoTalkPeer.cgi?action=attachState		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.250/cgi-bin/VideoTalkPeer.cgi?action=attachState			

Response Params (multipart ; key=value format in body)				
Name	Type	R/O	Description	Example
SID	integer	No	Server subscription ID	13606
state	object	No	Called state	
+State	string	No	Call state Enumchar[32]{ "Ringing": passive call "Inviting": active call "Answer" "Refuse" "Cancel" "Hangup" "Busying" }	Answer
+Talkback	object	No	Transmission method, audio and video encoding format, listening port and other session description information	
++Pack	string	No	Packing Mode It only supports "RTP" currently.	RTP
++Protocol	string	No	Transport protocol type Enumchar[32]{ TCP UDP }	UDP
++Type	string	No	Talk type	Talk

			Enumchar[32]{ "Talk": Talk "Broadcast": Broadcast }	
++Audio	object	No	Audio Description	
+++Format	object[]	No	Supported audio decoding formats	
++++Compression	string	No	Audio compression format Enumchar[32]{ PCM ADPCM "G.711A" "G.711Mu" "G.726" "G.729" "MPEG2" AMR AAC }	PCM
++++Frequency	integer	No	Audio Sampling Frequency	44000
++++Depth	integer	No	Sampling depth	16
++Video	object	No	Video description	
+++VideoPort	integer	No	Receive video stream port When receiving multiple video channels, use different ports respectively.	7000
+++Format	object[]	No	Supported video decoding formats	
++++Compression	string	No	Video compression format Enumchar[32]{ "MPEG4" "MPEG2" "MPEG1" "MJPG" "H.263" "H.264" }	H.264
++++Frequency	integer	No	Video sampling frequency	9000
++MediaAddr	string	No	Streaming address It can be an intercom or broadcast address	224.10.10.10

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: closed

--<boundary>

Content-Type: text/plain

Content-Length: <data length>

```
SID=315
state.State=Answer
state.Talkback.Pack=RTP
state.Talkback.Protocol=UDP
state.Talkback.Type=Talk
state.Talkback.Audio.AudioPort=6000
state.Talkback.Audio.Format[0].Compression=PCM
state.Talkback.Audio.Format[0].Frequency=44000
state.Talkback.Audio.Format[0].Depth=16
state.Talkback.Audio.Format[1].Compression=G.711A
state.Talkback.Audio.Format[1].Frequency=44000
state.Talkback.Audio.Format[1].Depth=16
state.Talkback.Video.VideoPort=7000
state.Talkback.Video.Format[0].Compression=H.264
state.Talkback.Video.Format[0].Frequency=90000
state.Talkback.Video.Format[1].Compression=MJPEG
...
state.Talkback.MediaAddr=224.10.10.10
--<boundary>
...
```

13.1.2 Unsubscribe Video Talk Status

Unsubscribe the video talk status.

Request URL	http://<server>/cgi-bin/VideoTalkPeer.cgi?action=detachState			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
SID	int	R	the subscribe id, which is the response of attachState	13606
Request Example				
http://192.168.1.250/cgi-bin/VideoTalkPeer.cgi?action=detachState&SID=101				

Response Params (OK in body)

Response Example

OK

13.1.3 Invite Server on Video Talk

Start the video talk conversation.

Request URL	http://<server>/cgi-bin/VideoTalkPeer.cgi?action=invite			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example

Talkback	integer	R	Channel number, starting from 1	1
+Pack	string	O	Packing Mode It only supports "RTP" currently.	RTP
+Protocol	string	O	Transport protocol type Enumchar[32]{ TCP "UDP" }	UDP
+Type	string	O	Talk type Enumchar[32]{ "Talk": Talk "Broadcast": Broadcast }	Talk
+Audio	object	O	Audio Description	
++Format	object[]	O	Supported audio decoding formats	
+++Compression	string	O	Audio compression format Enumchar[32]{ PCM "ADPCM" "G.711A" "G.711Mu" "G.726" "G.729" "MPEG2" "AMR" "AAC" }	PCM
+++Frequency	integer	O	Audio Sampling Frequency	44000
+++Depth	integer	O	Sampling depth	16
+Video	object	O	Video description	
++VideoPort	integer	O	Receive video stream port When receiving multiple video channels, use different ports respectively.	7000
++Format	object[]	O	Supported video decoding formats	
+++Compression	string	O	Video compression format Enumchar[32]{ "MPEG4" "MPEG2" "MPEG1" "MJPG" "H.263" "H.264" }	H.264
+++Frequency	integer	O	Video sampling frequency	9000
+MediaAddr	string	O	Streaming address It can be an intercom or broadcast address	224.10.10.10

Request Example

```
http://192.168.1.250/cgi-bin/VideoTalkPeer.cgi?action=invite&Talkback.Protocol=UDP&Talkback.Type=Talk&Talkback.MediaAddr=224.10.10.10
```

Response Params (OK in body)**Response Example**

OK

13.1.4 Cancel the Video Talk

Cancel video talk conversation.

Request URL	http://<server>/cgi-bin/VideoTalkPeer.cgi?action=cancel		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.250/cgi-bin/VideoTalkPeer.cgi?action=cancel			

Response Params (OK in body)**Response Example**

OK

13.1.5 Answer the Invitation

Answer the call.

Request URL	http://<server>/cgi-bin/VideoTalkPeer.cgi?action=answer		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Talkback	integer	R	Channel number, starting from 1
+Pack	string	O	Packing Mode It only supports "RTP" currently.
+Protocol	string	O	Transport protocol type Enumchar[32]{ TCP "UDP" }
+Type	string	O	Talk type Enumchar[32]{ "Talk": Talk "Broadcast": Broadcast }
+Audio	object	O	Audio Description
++Format	object[]	O	Supported audio decoding formats
+++Compression	string	O	Audio compression format
			PCM

			Enumchar[32]{ PCM "ADPCM" "G.711A" "G.711Mu" "G.726" "G.729" "MPEG2" "AMR" "AAC" }	
+++Frequency	integer	O	Audio Sampling Frequency	44000
+++Depth	integer	O	Sampling depth	16
+Video	object	O	Video description	
++VideoPort	integer	O	Receive video stream port When receiving multiple video channels, use different ports respectively.	7000
++Format	object[]	O	Supported video decoding formats	
+++Compression	string	O	Video compression format Enumchar[32]{ "MPEG4" "MPEG2" "MPEG1" "MJPEG" "H.263" "H.264" }	H.264
+++Frequency	integer	O	Video sampling frequency	9000
+MediaAddr	string	O	Streaming address It can be an intercom or broadcast address	224.10.10.10

Example

http://192.168.1.250/cgi-bin/VideoTalkPeer.cgi?action=answer&Talkback.Protocol=UDP&Talkback.Type=Talk&Talkback.MediaAddr=224.10.10.10

Response Params (OK in body)
Response Example
OK

13.1.6 Refuse to Answer the Video Talk Invitation

Refuse answer the call.

Request URL	http://<server>/cgi-bin/VideoTalkPeer.cgi?action=refuse
Method	GET
Request Params (key=value format in URL)	

Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.250/cgi-bin/VideoTalkPeer.cgi?action=refuse				

Response Params (OK in body)

Response Example

OK

13.1.7 Hang Up

Close it when the conversation is over.

Request URL	http://<server>/cgi-bin/VideoTalkPeer.cgi?action=hangup		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.250/cgi-bin/VideoTalkPeer.cgi?action=hangup			

Response Params (OK in body)

Response Example

OK

13.2 Intelligent Building Record

13.2.1 Query Video Talk Log Record

This API is supported by VTO products.

Table 13-1

Syntax	http://<server>/cgi-bin/recordFinder.cgi?action=find&name=VideoTalkLog[&condition.CallType=<Type>&condition.EndState=<State>&count=<countNo>]
Method	GET
Description	Find the VideoTalkLog record.
Example	http://192.168.1.250/cgi-bin/recordFinder.cgi?action=find&name=VideoTalkLog&condition.CallType=Incoming&condion.EndState=Missed&count=500
Success Return	totalCount =1000 found =500 records[0].ReINDo=789 records[0].CreateTime=123456789 records[0]. CallType =Incoming records[0]. EndState =Received records[0].PeerNumber=501 ...
Comment	Parameters in URL:

	<p>Type: call type State: end state of the call countNo: the number of records to get</p> <p>Parameters in Response : totalCount : the record count which match condition found : the record count to return CallType: call type. The range is {"Incoming", "Outgoing"}. EndState: the range is {"Missed" , "Received"}</p>
--	--

13.2.2 Insert Announcement Record

This api is supported by VTO products.

Table 13-2

Syntax	http://<server>/cgi-bin/recordUpdater.cgi?action=insert&name=Announcement&Content=< Content >&ExpirTime=< ExpirTime >&IssueTime=< IssueTime >&Title=< Title >&User=< User >&State=< State >&ReadFlag=< ReadFlag >
Method	GET
Description	Insert the Announcement record.
Example	http://192.168.1.250/cgi-bin/recordUpdater.cgi?action=insert&name=Announcement&Content=stringData&ExpirTime=2012-01-01%2012:00:00&IssueTime=2012-01-01%2012:00:00&Title=Anounce1&User=101&State=0&ReadFlag=0
Success Return	ReINDo=< ReINDo >
Comment	<p>Parameters in URL: Content: Announcement Content ExpirTime: the time when the Announcement expire, format: 2012-01-01%2012:00:00 IssueTime: Announcement issue time, format: 2012-01-01%2012:00:00 Title: title of the announcement User: the number the Announcement issued to State: the state of the Announcement. 0 init, 1 send , 2 overdue ReadFlag: the read flag. 0 not read, 1 read.</p> <p>Parameters in Response : ReINDo: the record index of the new record</p>

13.2.3 Query Alarm Record

Table 13-3

Syntax	http://<server>/cgi-bin/recordFinder.cgi?action=find&name=AlarmRecord[&StartTime=< startTime >&EndTime=< endTime >&count=< countNo >]
Method	GET
Description	Find the AlarmRecord record. (This api is supported by video talk device.)
Example	http://192.168.1.250/cgi-bin/recordFinder.cgi?action=find&name=AlarmRecord&StartTime=2014-8-25%2000:02:32&EndTime=2014-8-25%2001:02:32&count=500

Success Return	totalCount =1000 found =500 records[0].ReINDo=789 records[0].CreateTime=123456789 records[0].Channel=0 records[0]. SenseMethod =DoorMagnetism records[0].RoomNumber=501 records[0].ReadFlag=0 records[0]. Comment =Friend ...
Comment	<p>Parameters in URL:</p> <p>startTime: The start time ,format : 2014-8-25%2000:01:32</p> <p>endTime: The end time, format: 2014-8-25%2000:02:32</p> <p>countNo: the number of records to get, The record count, default 1024</p> <p>Parameters in Response :</p> <p>totalCount : the record count which match condition</p> <p>found : the record count to return</p> <p>SenseMethod :the range is { "DoorMagnetism", "PassiveInfrared", "GasSensor", "SmokingSensor", "WaterSensor", "ActiveInfrared", "CallButton", "UrgencyButton", "Steal", "Perimeter", "PreventRemove", "DoorBell" }</p>

13.3 SIP

The user needs to connect video intercom through CGI protocol, but do not know how to use this protocol to configure the device. This document provides the corresponding CGI protocol according to the configuration data format involved in the specific service.

13.3.1 [Config] SIP Configuration

- Get SIP Configuration

Table 13-4

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=SIP					
Method	GET					
Description	Get SIP server configuration.					
[Request Params] (None)						
[Response Params] (key=value format)						
Name	Type	R/O	Param Description			
table	object	R	Configuration data table			
+SIP	object	R	Configuration name			
++AccoutName	string	O	Account name			
++SIPServer	string	R	SIP server IP address or domain name			
++SIPServerPort	int	R	SIP server port number			

++OutboundProxy	string	O	Proxy server IP address or domain name
++OutboundProxyID	string	O	Proxy server ID
++OutboundProxyPort	Int	O	Proxy server port number
++UserID	String	R	User account ID Generally a phone number
++UserType	int	O	User type
++AuthID	String	R	Authentication ID
++AuthPassword	String	R	Authentication password
++STUNServer	String	O	STUN (Simple Traversal of UDP over NATs) server IP address or domain name
++RegisterRealm	String	R	Registration domain
++RegExpiration	int	O	Registration interval, in seconds
++LocalSIPPort	Int	R	Local SIP port 0-65535
++LocalRTPPort	int	R	Local RTP port 0-65535
++UnregisterOnReboot	bool	O	Restart to delete registration information. true: Delete false: Not delete
++DefaultCallNumber	string	O	Default call number
++MediaDetail	Object	O	Media configuration
+++VideoStream	string	O	Video stream enumchar[32]{ "Main": Main stream "Extra1": Sub stream 1 "Extra2": Sub stream 2 "Extra3": Sub stream 3 }
+++AudioStream	string	O	Audio stream enumchar[32]{ "Main": Main stream "Extra1": Sub stream 1 "Extra2": Sub stream 2 "Extra3": Sub stream 3 }
++RouteEnable	bool	O	Enable SIP cross-router or not.
++Route	string	O	Router address, which can be IP address or domain name.
++SIPServerLoginUserName	string	O	Username to log in to VTNC For intelligent building only
++SIPServerLoginPWD	string	O	Password to log in to VTNC For intelligent building only
++IsMainVTO	string	O	Whether the door station is a standby server For intelligent building only

++SIPServerRedundancy	string	O	Standby server IP address For intelligent building only
++SIPServerRedundancyUserName	string	O	Standby server login username For intelligent building only
++SIPServerRedundancyPassWord	string	O	Standby server login password For intelligent building only
++AnalogNumberStart	string	O	The start number of the analog indoor monitor supported in the analog system For intelligent building only
++AnalogNumberEnd	string	O	The end number of the analog indoor monitor supported in the analog system For intelligent building only
++UserEnable	bool	R	Enable registration For intelligent building only true: Register to the SIP server. false: Not register to the SIP server.

[Example]

Request	http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=SIP
Response	table.SIP.AuthID=8001 table.SIP.AuthPassword=pass123456 table.SIP.IsMainVTO=0 table.SIP.LocalRTTPort=15000 table.SIP.LocalSIPPort=5060 table.SIP.OutboundProxy=192.168.1.111 table.SIP.OutboundProxyID=8000 table.SIP.OutboundProxyPort=5060 table.SIP.RegisterRealm=VDP table.SIP.Route[0]=sip:10.30.1.2:5060;lr table.SIP.RouteEnable=true table.SIP.SIPServer=192.168.1.111 table.SIP.SIPServerID=8000 table.SIP.SIPServerLoginPWD=pass123456 table.SIP.SIPServerLoginUserName=admin table.SIP.SIPServerPort=5060 table.SIP.SIPServerRedundancyPassWord=pass123456 table.SIP.SIPServerRedundancyUserName=admin table.SIP.UserEnable=true table.SIP.UserID=8001 table.SIP.UserType=2

- Set SIP Configuration

Table 13-5

URL	http://<server>/cgi-bin/configManager.cgi?action=setConfig
Method	GET
Description	Modify SIP configuration.
[Request Params] (key=value at URL)	

Name	Type	R/O	Param Description
+SIP	object	R	Configuration name
++AccoutName	string	O	Account name
++SIPServer	string	R	SIP server IP address or domain name
++SIPServerPort	int	R	SIP server port number
++OutboundProxy	string	O	Proxy server IP address or domain name
++OutboundProxyID	string	O	Proxy server ID
++OutboundProxyPort	Int	O	Proxy server port number
++UserID	String	R	User account ID Generally a phone number
++UserType	int	O	User type
++AuthID	String	R	Authentication ID
++AuthPassword	String	R	Authentication password
++STUNServer	String	O	STUN (Simple Traversal of UDP over NATs) server IP address or domain name
++RegisterRealm	String	R	Registration domain
++RegExpiration	int	O	Registration interval, in seconds
++LocalSIPPort	Int	R	Local SIP port 0-65535
++LocalRTPPort	int	R	Local RTP port 0-65535
++UnregisterOnReboot	bool	O	Restart to delete registration information. true: Delete false: Not delete
++DefaultCallNumber	string	O	Default call number
++MediaDetail	Object	O	Media configuration
+++VideoStream	string	O	Video stream enumchar[32]{ "Main": Main stream "Extra1": Sub stream 1 "Extra2": Sub stream 2 "Extra3": Sub stream 3 }

+++AudioStream	string	O	Audio stream enumchar[32]{ "Main": Main stream "Extra1": Sub stream 1 "Extra2": Sub stream 2 "Extra3": Sub stream 3 }
++RouteEnable	bool	O	Enable SIP cross-router or not.
++Route	string	O	Router address, which can be IP address or domain name.
++SIPServerLog inUserName	string	O	Username to log in to VTNC For intelligent building only
++SIPServerLog inPWD	string	O	Password to log in to VTNC For intelligent building only
++IsMainVTO	string	O	Whether the door station is a standby server For intelligent building only
++SIPServerRe dundancy	string	O	Standby server IP address For intelligent building only
++SIPServerRe dundancyUserN ame	string	O	Standby server login username For intelligent building only
++SIPServerRe dundancyPassW ord	string	O	Standby server login password For intelligent building only
++AnalogNumbe rStart	string	O	The start number of the analog indoor monitor supported in the analog system For intelligent building only
++AnalogNumbe rEnd	string	O	The end number of the analog indoor monitor supported in the analog system For intelligent building only
++UserEnable	bool	R	Enable registration For intelligent building true: Register to the SIP server. false: Not register to the SIP server.
[Response Params] (OK)			
[Example]			
Request	http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&SIP.SIPServer=192.168.1.108&SIP.SIPServerPort=5060&SIP.RegisterRealm=VDP&SIP.RouteEnable=true		
Response	OK		

13.3.2 [Config] Registrar Configuration

- Get Registrar Configuration

Table 13-6

Syntax	<a href="http://<server>/cgi-bin/configManager.cgi?action=getConfig&name= Registrar">http://<server>/cgi-bin/configManager.cgi?action=getConfig&name= Registrar
--------	---

Method	GET					
Description	Get Registrar configuration.					
[Request Params] (None)						
[Response Params] (key=value format)						
Name	Type	R/O	Param Description			
table	object	R	Configuration data table			
+Registrar	object[]	R	One-dimensional array. Each subscript represents a registration server.			
++RegistrarName	char[]	O	Registration server name, which must be unique. (Note: For VTH, use the IP address and port in the configuration to connect VTO.)			
++Enable	bool	O	Register to the registration server or not.			
++ServerType	enumchar[32]	R	Server type, which is used only for SIP video talk, instead of VT video talk. Enumchar[32]{ "VTO" "H500" "VTNC" "ZYCOO" "ThirdParty" "3CXSystem" "Asterisk" }			
++GeneralServerInfo	object	O				
+++Address	char[40]	O	IP address			
+++Port	uint	O	Registration port number			
+++Password	char[64]	O	Password registered to the register server, which will be used when registration authentication is required.			
[Example]						
Request	http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=Registrar					
Response	table.Registar[0].ServerType=VTO					

- Set Registrar Configuration

Table 13-7

URL	http://<server>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Modify Registrar configuration.		
[Request Params] (key=value at URL)			
Name	Type	R/O	Param Description
+Registrar	object[]	R	One-dimensional array. Each subscript represents a register server.
++RegistrarName	char[]	O	Registration server name, which must be unique. (Note: For VTH, use the IP address and port in the configuration to connect VTO.)
++Enable	bool	O	Register to the registration server or not.

++ServerType	enumchar[32]	R	Server type, which is used only for SIP video talk, instead of VT video talk. Enumchar[32]{ "VTO" "H500" "VTNC" "ZYCOO" "ThirdParty" "3CXSystem" "Asterisk" }
++GeneralServerInfo	object	O	
+++Address	char[40]	O	IP address
+++Port	uint	O	Registration port number
+++Password	char[64]	O	Password registered to the registration server, which will be used when registration authentication is required.
[Response Params] (OK)			
[Example]			
Request	http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&Registar.ServerType=VTO		
Response	OK		

13.4 Room Number Database Management

13.4.1 Adding Room Number

Table 13-8

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=insert&name=VideoTalkContact		
Method	GET		
Description	Add room number to the contacts.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
CreateTime	uint	O	Start time UTC seconds, read only
FirstName	string	O	First name
FamilyName	string	O	Last name
VTShortNumber	string	R	Short number for video intercom
VTMiddleNumber	string	O	Middle number for video intercom
VTLongNumber	string	O	Long number for video intercom or serial number of analog indoor monitor
VTNetAddress	string	O	Network address for video intercom
MacAdress	string	O	MAC address
VTOPosition	string	O	Door number linked with indoor monitor
VTSlaveBindMod	enumint	O	Mode when accessing to the analog indoor monitor for video

e			talk Enumint{ 0: SubAddress 1: SubID+SubPort }
VTSlaveld	uint32	O	Allocator address when accessing to the analog indoor monitor for video talk (Change string to uint32 for consistency)
VTSlavePort	uint32	O	Allocator port when accessing to the analog indoor monitor for video talk (Change string to uint32 for consistency)
VTSlaveAddress	string	O	Address of the analog indoor monitor for video talk
NickName	string	O	Nickname
Notes	string	O	Notes
Type	enumchar[32]	O	User type Enumchar[32]{ "VTH": Indoor monitor "VTO": Door station }
.RegisterType	enumchar[32]	R	Registration method Enumchar[32]{ "public" "local" }
VTHPassword	string	R	Registration password
VTOBuilding	string	O	Building number
VTOUnit	string	O	Unit number
GroupNumber	string	O	Group
Channel	uint32	O	Channel number, based on which the mobile phone subscribes to the call notification message.
Floors	char[256] [4]	O	Floor number (lift control requirements), with up to 256 characters
LiftControlByVTH	bool	O	Lift control is triggered by the indoor monitor opening the door.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
RelINDo	int	R	Record ID of successfully creation
[Example]			
Request	http://<server>/cgi-bin/recordUpdater.cgi?action=insert&name=VideoTalkContact&VTShort Number=101&RegisterType=public&VTHPassword=pass123456		
Response	RelINDo=12345		

13.4.2 Getting Records by Video Talk short number

Table 13-9

URL	http://<server>/cgi-bin/recordFinder.cgi?action=find&name=VideoTalkContact
-----	--

Method	GET		
Description	Search for records by room number		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
count	int	O	Maximum record number of search result
condition	object	R	search condition
+VTShortNumber	string	R	video talk short number
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
totalCount	int	O	total record number of search result
found	int	O	Number of records returned
records	array<object>	R	Records returned
+RelDo	int	R	Record ID
+CreateTime	uint	O	Start time UTC seconds, read only
+FirstName	string	O	First name
+FamilyName	string	O	Last name
+VTShortNumber	string	R	Short number for video intercom
+VTMiddleNumber	string	O	Middle number for video intercom
+VTLONGNumber	string	O	Long number for video intercom or serial number of VTH analog indoor monitor
+VTNetAddress	string	O	Network address for video intercom
+MacAdress	string	O	MAC address
+VTOPosition	string	O	Door number linked with indoor monitor
+VTSIaveBindMode	enumint	O	Mode when accessing to the analog indoor monitor for video talk Enumint{ 0: SubAddress 1: SubID+SubPort }
+VTSlaveld	uint32	O	Allocator address when accessing to the analog indoor monitor for video talk (Change string to uint32 for consistency)
+VTSIavePort	uint32	O	Allocator port when accessing to the analog indoor monitor for video talk (Change string to uint32 for consistency)
+VTSIaveAddres	string	O	Address of the analog indoor monitor for video talk
+NickName	string	O	Nickname
+Notes	string	O	Notes
+Type	enumchar[32]	O	User type Enumchar[32]{

			"VTH": Indoor monitor "VTO": Door station }
+RegisterType	enumchar[32]	R	Registration method Enumchar[32]{ "public" "local" }
+VTHPassword	string	R	Registration password
+VTOBuilding	string	O	Building number
+VTOUnit	string	O	Unit number
+GroupNumber	string	O	Group
[Example]			
Request	http://<server>/cgi-bin/recordFinder.cgi?action=find&name=VideoTalkContact&count=10&condition.VTShortNumber=101		
Response	totalCount=10 found=1 records[0].ReINDo=12345 records[0].CreateTime=123456789 records[0].FirstName=San records[0].FamilyName=Zhang records[0].VTShortNumber=0101 records[0].VTMiddleNumber=11010101 records[0].VTLONGNumber=330103001101010151 records[0].VTNetAddress=127.0.0.1 records[0].MacAdress=0A:3E:FF:2A:50:41 records[0].VTOPosition=01018001 records[0].VTSlaveBindMode=0 records[0].VTSlaveId=1258421 records[0].VTSlavePort=1258421 records[0].VTSlaveAddress=04:b3:01:f7 records[0].NickName=Nick records[0].Notes=Friend records[0].Type=VTH records[0].RegisterType=public records[0].VTHPassword=123456 records[0].VTOBuilding=01 records[0].VTOUnit=01 records[0].GroupNumber=301 records[0].Channel=1 records[0].Floors[0]=1 records[0].LiftControlByVTH=true		

13.4.3 Getting Records by reINDo

Table 13-10

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=get&name=VideoTalkContact		
Method	GET		
Description	Search for room number records by reINDo.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
reINDo	int	R	Record ID
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
+ReINDo	int	R	Record ID
+CreateTime	uint	O	Start time UTC seconds, read only
+FirstName	string	O	First name
+FamilyName	string	O	Last name
+VTShortNumber	string	R	Short number for video intercom
+VTMiddleNumber	string	O	Middle number for video intercom
+VTLONGNumber	string	O	Long number for video intercom or serial number of VTH analog indoor monitor
+VTNetAddress	string	O	Network address for video intercom
+MacAdress	string	O	MAC address
+VTOPosition	string	O	Door number linked with indoor monitor
+VTSlaveBindMode	enumint	O	Mode when accessing to the analog indoor monitor for video talk Enumint{ 0: SubAddress 1: SubID+SubPort }
+VTSlaveId	uint32	O	Allocator address when accessing to the analog indoor monitor for video talk (Change string to uint32 for consistency)
+VTSlavePort	uint32	O	Allocator port when accessing to the analog indoor monitor for video talk (Change string to uint32 for consistency)
+VTSlaveAddress	string	O	Address of the analog indoor monitor for video talk
+NickName	string	O	Nickname
+Notes	string	O	Notes
+Type	enumchar[32]	O	User type Enumchar[32]{ "VTH": Indoor monitor "VTO": Door station }
+RegisterType	enumchar[32]	R	Registration method Enumchar[32]{ "public"

			"local" }
+VTHPassword	string	R	Registration password
+VTOBuilding	string	O	Building number
+VTOUnit	string	O	Unit number
+GroupNumber	string	O	Group
[Example]			
Request	http://<server>/cgi-bin/recordUpdater.cgi?action=get&name=VideoTalkContact&relINDo=12345		
Response	record.RelINDo=12345 record.CreateTime=123456789 record.FirstName=Dafei record.FamilyName=Wang record.VTShortNumber=0101 record.VTMiddleNumber=11010101 record.VTLongNumber=330103001101010151 record.VTNetAddress=127.0.0.1 record.MacAdress=0A:3E:FF:2A:50:41 record.VTOPosition=01018001 record.VTSlaveBindMode=0 record.VTSlaveId=1258421 record.VTSlavePort=1258421 record.VTSlaveAddress=04:b3:01:f7 record.NickName=Nick record.Notes=Friend record.Type=VTH record.RegisterType=public record.VTHPassword=123456 record.VTOBuilding=01 record.VTOUnit=01 record.GroupNumber=301 record.Channel=1 record.Floors[0]=1 record.LiftControlByVTH=true		

13.4.4 Updating Room Number Records

Table 13-11

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=update&name=VideoTalkContact		
Method	GET		
Description	Update Room number database records.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
relINDo	int	R	Record ID
CreateTime	uint	O	Start time UTC seconds, read only

FirstName	string	O	First name
FamilyName	string	O	Last name
VTShortNumber	string	O	Short number for video intercom
VTMiddleNumber	string	O	Middle number for video intercom
VTLongNumber	string	O	Long number for video intercom or serial number of VTH analog indoor monitor
VTNetAddress	string	O	Network address for video intercom
MacAdress	string	O	MAC address
VTOPosition	string	O	Door number linked with indoor monitor
VTSlaveBindMode	enumint	O	Mode when accessing to the analog indoor monitor for video talk Enumint{ 0: SubAddress 1: SubID+SubPort }
VTSlaveld	uint32	O	Allocator address when accessing to the analog indoor monitor for video talk (Change string to uint32 for consistency)
VTSlavePort	uint32	O	Allocator port when accessing to the analog indoor monitor for video talk (Change string to uint32 for consistency)
VTSlaveAddress	string	O	Address of the analog indoor monitor for video talk
NickName	string	O	Nickname
Notes	string	O	Notes
Type	enumchar[32]	O	User type Enumchar[32]{ "VTH": Indoor monitor "VTO": Door station }
.RegisterType	enumchar[32]	O	Registration method Enumchar[32]{ "public" "local" }
VTHPassword	string	O	Registration password
VTOBuilding	string	O	Building number
VTOUnit	string	O	Unit number
GroupNumber	string	O	Group
Channel	uint32	O	Channel number, based on which the mobile phone subscribes to the call notification message.
Floors	char[256][4]	O	Floor number (lift control requirements), with up to 256 characters
LiftControlByVTH	bool	O	Lift control is triggered by the indoor monitor opening the door.
[Response Params] (OK)			

[Example]	
Request	http://<server>/cgi-bin/recordUpdater.cgi?action=update&name=VideoTalkContact&reINDo=12345&FirstName=Green&FamilyName=Jim&VTHPassword=654321
Response	OK

13.4.5 Deleting Records by reINDo

Table 13-12

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=remove&name=VideoTalkContact		
Method	GET		
Description	Delete records by reINDo.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
reINDo	int	R	Record ID
[Response Params] (OK)			
[Example]			
Request	http://<server>/cgi-bin/recordUpdater.cgi?action=remove&name=VideoTalkContact&reINDo=12345		
Response	OK		

13.4.6 Clearing All Room Numbers

Table 13-13

URL	http://<server>/cgi-bin/recordUpdater.cgi?action=clear&name=VideoTalkContact		
Method	GET		
Description	Delete all records .		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Database name, which must be VideoTalkContact.
[Response Params] (OK)			
[Example]			
Request	http://<server>/cgi-bin/recordUpdater.cgi?action=clear&name=VideoTalkContact		
Response	OK		

13.4.7 Getting Total Quantity of Room Number

Table 13-14

URL	http://<server>/cgi-bin/recordFinder.cgi?action=getQuerySize&name=VideoTalkContact		
Method	GET		
Description	Get the total number of records of the database VideoTalkContact.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Database name, which must be VideoTalkContact.

[Response Params] (key=value format)			
Name	Type	R/O	Param Description
count	int	R	Total number of records
[Example]			
Request	http://<server>/cgi-bin/recordFinder.cgi?action=getQuerySize&name=VideoTalkContact		
Response	Count=200		

13.5 ElevatorFloorCounter

13.5.1 Set Elevator Floor Info

URL	http://<server>/cgi-bin/ElevatorFloorCounter.cgi?action=setElevatorFloorInfo&CheckBaseFloor=<Second>&channel=<ChannelNo>		
Method	GET		
Description	Set elevator floor infomation		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
Channel	int	R	ChannelNo, start from 1, default 1
CheckBaseFloor	string	R	Calibrate base floor, unit floor, It's not necessarily numbers
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
[Example]			
Request	http://192.168.1.250/cgi-bin/ElevatorFloorCounter.cgi?action=setElevatorFloorInfo&CheckBaseFloor=5&Channel=1		
Response	OK		

13.5.2 Get Elevator WorkInfo

URL	http://<server>/cgi-bin/ElevatorFloorCounter.cgi?action=getElevatorWorkInfo&Channel=<ChannelNo>		
Method	GET		
Description	Get elevator work infomation		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
Channel	int	R	ChannelNo, start from 1, default 1
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
table	object	R	Operation information, including the current floor, status, speed, whether to calibrate
[Example]			
Request	http://192.168.1.250/cgi-bin/ElevatorFloorCounter.cgi?action=getElevatorWorkInfo&Channel=1		
Response	table.ElevatorFloor=20 table.ElevatorWorkState=1 table.ElevatorWorkSpeed=20		

table.IsCheckFloor=true

13.5.3 Get Capability

URL	http://<server>/cgi-bin/ElevatorFloorCounter.cgi?action=getCaps&Channel=<ChannelNo>		
Method	GET		
Description	Get elevator floor capacities.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
Channel	int	R	ChannelNo, start from 1, default 1
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
caps	object	R	elevator floor capacities
+IsElevatorFloorCounter	bool	R	Whether the floor configuration function is supported
[Example]			
Request	http://192.168.1.250/cgi-bin/ ElevatorFloorCounter.cgi?action=getCaps&channel=1		
Response	caps.IsElevatorFloorCounter=true		

14.1 File Finder

14.1.1 Create a File Finder

Table 14-1

Syntax	http://<server>/cgi-bin/FileFindHelper.cgi?action=startFind&condition.channel=< ChannelNo >&condition.startTime=< start >&condition.endTime=< end >&condition.streamType=< stream >[&condition.flags[0]=< flag >&condition.events[0]=< event >&combineMode.granularity=< granularityValue >&combineMode.types[0]=< combineType >]
Method	GET
Description	Start to find files.
Example	<p>Find a file in channel 1. Event type is "AlarmLocal" or "VideoMotion". Time lines between 2014-1-1 12:00:00 and 2015-1-10 12:00:00. Combine "AlarmLocal" or "VideoMotion" files with granularity 16. URL is: http://172.23.1.66/cgi-bin/fileFindHelper.cgi?action=startFind&condition.channel=1&condition.startTime=2014-1-1%2012:00:00&condition.endTime=2015-1-10%2012:00:00&condition.streamType=Main&condition.flags[0]=Event&condition.events[0]=AlarmLocal&condition.events[1]=VideoMotion&combineMode.granularity=16&combineMode.types[0]=AlarmLocal&combineMode.types[0]=VideoMotion</p>
Success Return	result=08137
Comment	<p>Start to find a file with the above condition and combine files with certain type. If it succeeds, return to find id. Otherwise return to Error.</p> <p>Parameters in URL:</p> <p>ChannelNo: integer, video channel index which starts from 1.</p> <p>start / end: the start/end time when recording.</p> <p>flag: which flags of the file you want to find. It is an array. The index starts from 0. The range of flag is {"Timing", "Marked", "Event", "Restrict"}. If omitted, find files with all the flags.</p> <p>event: by which event the record file is triggered. It is an array. The index starts from 0. The range of event is {"AlarmLocal", "VideoMotion", "VideoLoss"}. This condition can be omitted. If omitted, find files of all the events.</p> <p>stream: which video stream type you want to find. The range of stream is {"Main", "Extra1", "Extra2", "Extra3"}.</p> <p>combineType: which types of the file you want to combined. It is an array. The index starts from 0. The range of combine type is {"AlarmLocal", "VideoMotion", "Timing", "VideoLoss"}. This condition can be omitted. If omitted, file will not be combined.</p> <p>granularityValue: by which granularity to combine files</p> <p>Example:</p>

```

File 1:
items[0].Channel =1
items[0].StartTime =2011-1-1 12:00:00
items[0].EndTime =2011-1-1 13:00:00
items[0].Events[0]=AlarmLocal
items[0].VideoStream=Main
items[0].Length =790
items[0].Duration = 3600

```

```

File 2:
items[0].Channel =1
items[0].StartTime =2011-1-1 13:00:00
items[0].EndTime =2011-1-1 14:00:00
items[0].Events[0]=AlarmLocal
items[0].VideoStream=Main
items[0].Length =790
items[0].Duration = 3600

```

file1 and file2 will be combined to file3

```

File 3:
items[0].Channel =1
items[0].StartTime =2011-1-1 12:00:00
items[0].EndTime =2011-1-1 14:00:00
items[0].Events[0]=AlarmLocal
items[0].VideoStream=Main
items[0].Length =1580
items[0].Duration = 7200

```

14.1.2 Create a Motion File Finder

Table 14-2

Syntax	<code>http://<server>/cgi-bin/FileFindHelper.cgi?action=startMotionFind&condition.channel=<channelNo>&condition.startTime=<start>&condition.endTime=<end>&condition.streamType=<stream>&motionRegion.senseLevel=<level>[&motionRegion.rects[rectNo][0]=<rect0>&motionRegion.rects[rectNo][1]=<rect1>&motionRegion.rects[rectNo][2]=<rect2>&motionRegion.rects[rectNo][3]=<rect3>]</code>
Method	GET
Description	Start to find motion files.
Example	<p>Find a file in channel 1, event type is "AlarmLocal" or "VideoMotion", and time between 2014-1-1 12:00:00 and 2015-1-10 12:00:00, motion region is [0,0,21,17] URL is:</p> <p><code>http://172.23.1.66/cgi-bin/fileFindHelper.cgi?action=startMotionFind&condition.channel=1&condition.startTime=2014-1-1%2012:00:00&condition.endTime=2015-1-10%2012:00:00&condition.streamType>Main&condition.flags[0]=Event&condition.events[0]=AlarmLocal&condition.events[1]=VideoMotio&motionRegion.senseLevel=1&motionRegion.rects[1][0]=0&motionRegion.rects[1][1]=0&motionRegion.rects[1][2]=21&motionRegion.rects[1][3]=17</code></p>

	3]=17
Success Return	result=08137
Comment	<p>Start to find a file with the above condition and combine files with certain type. If success, return find id, else return Error.</p> <p>Parameters in URL:</p> <ul style="list-style-type: none"> ChannelNo: integer, video channel index which starts from 1. start / end: the start/end time when recording. flag: which flags of the file you want to find. It is an array. The index starts from 0. The range of flag is {"Timing", "Marked", "Event", "Restrict"}. If omitted, find files with all the flags. event: by which event the record file is triggered. It is an array. The index starts from 0. The range of event is {"AlarmLocal", "VideoMotion"}. This condition can be omitted. If omitted, find files of all the events. stream: which video stream type you want to find. The range of stream is {"Main", "Extra1", "Extra2", "Extra3"}. level: the motion sensitive level, range is 0–6, 0 represent all level rectNo: the rects array index, start from 1 rect0 & rect1 & rect2 & rect3: relative coordinates, rect0 and rect2 range is 0—21, rect1 and rect3 range is 0—17. {0,0,0,0} top—left, {21,0,0,0} top—right, {0,17,0,0} bottom—left, {21,17,0,0} bottom—right

14.1.3 Get the File Information Found by the Finder

Table 14-3

Syntax	http://<server>/cgi-bin/FileFindHelper.cgi?action=findNext&findId=< findId >&count=< fileCount >
Method	GET
Description	Find the next files no more than <i>fileCount</i> number.
Example	http://192.168.1.250/cgi-bin/FileFindHelper.cgi?action=findNext&findId=08137&count=100
Success Return	found=1 items[0].channel =1 items[0].startTime =2011-1-1 12:00:00 items[0].endTime =2011-1-1 13:00:00 items[0].fileType =dav items[0].events[0]=AlarmLocal items[0].streamType>Main items[0].length =790 items[0].duration = 3600
Comment	findId : The find Id is created by API Create a file finder or API Create a motion file finder . Must create a finder before finding files.

14.1.4 Stop the Finder

Table 14-4

Syntax	<code>http://<server>/cgi-bin/FileFindHelper.cgi?action=stopFind&findId=<findId></code>
Method	GET
Description	Stop the searching operation.
Example	<code>http://192.168.1.250/cgi-bin/FileFindHelper.cgi?action=stopFind&findId=08137</code>
Success Return	OK
Comment	findId: The find Id is created by API Create a file finder or API Create a motion file finder . Must create a finder before finding files.

14.1.5 Get Bound Files

Table 14-5

Syntax	<code>http://<server>/cgi-bin/FileFindHelper.cgi?action=getBoundFile&condition.channel=<ChannelNo>&condition.startTime=<start>&condition.endTime=<end>&condition.streamType=<stream>[&condition.flags[0]=<flag>&condition.events[0]=<event>]</code>
Method	GET
Description	Get bound files.
Example	<code>http://<server>/cgi-bin/FileFindHelper.cgi?action=getBoundFile&condition.channel=1&condition.startTime=2014-1-1%2012:00:00&condition.endTime=2015-1-10%2012:00:00&condition.streamType>Main&condition.flags[0]=Timing</code>
Success Return	<pre>found=2 items[0].channel =1 items[0].startTime =2011-1-1 12:00:00 items[0].endTime =2011-1-1 13:00:00 items[0].flags [0]= Timing items[0].streamType>Main items[0].length =790 items[0].duration = 3600 items[1].channel =1 items[1].startTime =2011-1-1 13:00:00 items[1].endTime =2011-1-1 14:00:00 items[1].events[0]= Timing items[1].streamType>Main items[1].length =790 items[1].duration = 3600</pre>
Comment	Params is same as FileFindHelper.startFind

14.2 BandLimit

14.2.1 Get Bandwidth Limit State

Table 14-6

Syntax	http://<server>/cgi-bin/BandLimit.cgi?action=getLimitState
Method	GET
Description	Get bandwidth limit state.
Example	http://192.168.1.250/cgi-bin/bandLimit.cgi?action=getLimitState
Success Return	limit=true
Comment	—

14.3 Record Files Protection

14.3.1 Add Protection

Table 14-7

Syntax	http://<server>/cgi-bin/FileManager.cgi?action=addConditionList&condition.Types[0]=< paramValue >&condition.StartTime=< paramValue >&condition.EndTime=< paramValue >&condition.Channel[0]=< paramValue >
Method	GET
Description	Add protection or access control for record files.
Example	http://192.168.1.250/cgi-bin/FileManager.cgi?action=addConditionList&condition.Types[0]=RecordRestrict&condition.Types[1]=RecordProtect&condition.StartTime=2014-7-3%2021:02:32&condition.EndTime=2014-7-3%2023:02:32&condition.Channel[0]=1&condition.Channel[1]=3
Success Return	OK
Comment	In table below: TypeIndex : The index of type array ChIndex : The index of channel number array

Appendix:

ParamName	ParamValue type	Description
condition.Type[TypeIndex]	string	An array. The range is {"RecordProtect", "RecordRestrict"}.
condition.StartTime	string	The time format is "Y-M-D H-m-S", example 2011-7-3%2021:02:32
condition.EndTime	string	The time format is "Y-M-D H-m-S"
condition.Channel[ChIndex]	integer	Video channel index which starts from 1.

14.3.2 Cancel Protection

Table 14-8

Syntax	http://<server>/cgi-bin/FileManager.cgi?action=cancelConditionList&condition.Types[0]=<paramValue>&condition.StartTime=<paramValue>&condition.EndTime=<paramValue>&condition.Channel[0]=<paramValue>
Method	GET
Description	Cancel protection of record files.
Example	http://192.168.1.250/cgi-bin/FileManager.cgi?action=cancelConditionList&condition.Types[0]=RecordRestrict&condition.Types[1]=RecordProtect&condition.StartTime=2014-7-3%2021:02:32&condition.EndTime=2014-7-3%2023:02:32&condition.Channel[0]=1&condition.Channel[1]=3
Success Return	OK
Comment	paramValue as Appendix above.

14.3.3 Remove Protection

Table 14-9

Syntax	http://<server>/cgi-bin/FileManager.cgi?action=removeConditionList&condition.Types[0]=<paramValue>&condition.StartTime=<paramValue>&condition.EndTime=<paramValue>&condition.Channel[0]=<paramValue>
Method	GET
Description	Remove protection of record files.
Example	http://192.168.1.250/cgi-bin/FileManager.cgi?action=removeConditionList&condition.Types[0]=RecordRestrict&condition.Types[1]=RecordProtect&condition.StartTime=2014-7-3%2021:02:32&condition.EndTime=2014-7-3%2023:02:32&condition.Channel[0]=0&condition.Channel[1]=3
Success Return	OK
Comment	paramValue as Appendix above.

14.3.4 DownloadFile

download file

Request URL	http://<server>/cgi-bin/FileManager.cgi?action=downloadFile		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
fileName	string	R	filename or path
Request Example			Example
http://192.168.1.250/cgi-bin/FileManager.cgi?action=downloadFile&fileName=download.jpg			download.jpg

Response Params (multipart binary data)				
Name	Type	R/O	Description	Example
Response Example				
HTTP/1.1 200 OK				
Content-type: text/plain; charset=utf-8				
CONNECTION: close				
Set-Cookie:secure; HttpOnly				
CONTENT-LENGTH: <length>				
<Binary Data>				

14.3.5 UploadFile

Upload radar photos

Request URL	http://<server>/cgi-bin/FileManager.cgi?action=uploadFile&fileName=< <i>FileName</i> >&Path=< <i>Path</i> >			
Method	POST			
Request Params (key=value format in url)				
Name	Type	R/O	Description	Example
fileName	string	R	File name, only the types bmp and jpg are available for radars.	"xxxxxx.bmp"
Path	string	R	The file path to store	/upload_pic
Request Example				
POST http://192.168.1.250/cgi-bin/FileManager.cgi?action=uploadFile&fileName=xxxxxx.bmp&fileLength=124879 HTTP/1.1 Host: 192.168.1.250 Connection: keep-alive Content-Type: multipart/form-data;boundary=-----8655433224198 Content-Length: xxxxxxxx				
-----8655433224198 Content-Disposition:form-data;name="upload"; filename="xxxxxx.bmp" Content-Type: image/jpeg or application/x-MS-bmp				
photo data....				
-----8655433224198--				

Response Params (OK in body)				
Parameter	Type	Required	Description	Example
Response Example				
OK				

14.3.6 List all elements in the specified directory

Request URL	http://<server>/cgi-bin/api/FileManager/list			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
path	char[260]	R	To get the path, there will be internal security verification. For files uploaded externally, such as obtaining the SFTP directory, you can obtain the NAS configuration and find the corresponding directory for query.	"/mnt/dvr/sda0/2010/8/11/dav"
Request Example				
{ "path": "/mnt/dvr/sda0/2010/8/11/dav" }				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
elementInfo	object[]	R	File element information	
+type	enumchar[16]	O	File element type enumchar[16]{ "File" "Directory" }	"File"
+file	object	O	File info (valid when Type="File")	
++FileType	char[64]	O	file node type "Wireshark/tcpdump", See Wireshark packet capture file type	"Wireshark/tcpdump"
++CreateTime	char[20]	O	Creation time	"2010-4-15 9:58:32"
++ModifyTime	char[20]	O	Modification time	"2010-4-15 9:58:32"
++Size	double	O	File size, the decimal part is meaningless. Unit: byte	1873.0
++path	char[260]	O	Absolute path	"/mnt/dvr/sda0/2010/8/11/dav.jpg"
++Desc	char[128]	O	File Custom Description	"xxxxxx"
+directory	object	O	directory information ,it works when Type="Directory"	
++CreateTime	char[20]	O	Creation time, formatted as "Y-M-D H-m-S"	"2010-4-15 9:58:32"
++path	char[260]	O	Absolute path	"/mnt/dvr/sda0/2010/8/11/dav.jpg"

				1/dav"
Response Example				
{				

```
{
  "elementInfo": [
    {
      "type": "File",
      "file": {
        "FileType": "Wireshark/tcpdump",
        "CreateTime": "2010-4-15 9:58:32",
        "ModifyTime": "2010-4-15 9:58:32",
        "Size": 1873.0,
        "path": "/mnt/dvr/sda0/2010/8/11/dav.jpg",
        "Desc": "xxxxxx"
      },
      "directory": {
        "CreateTime": "2010-4-15 9:58:32",
        "path": "/mnt/dvr/sda0/2010/8/11/dav"
      }
    },...{}
  }
}
```

14.4 Daylight

14.4.1 Get Daylight

Table 14-10

Syntax	http://<server>/cgi-bin/global.cgi?action=getDST
Method	GET
Description	Get daylight saving time state.
Example	http://192.168.1.250/cgi-bin/global.cgi?action=getDST
Success Return	result = 1
Comment	result: 1/0, yes or not in daylight saving time

15. Other API

15.1 Discover Devices

15.1.1 Discover Devices on Internet

Table 15-1

Syntax	http://<server>/cgi-bin/deviceDiscovery.cgi?action=attach[&DeviceClass=< deviceClass >]
Method	GET
Description	Discover devices on internet.
Example	http://192.168.1.250/cgi-bin/deviceDiscovery.cgi?action=attach&DeviceClass=VTO
Success Return	deviceInfo[index].AlarmInputChannels=8 deviceInfo[index].AlarmOutputChannels=0 deviceInfo[index].DeviceClass=VTO deviceInfo[index].DeviceType=VTO2000A deviceInfo[index].HttpPort=80 deviceInfo[index].IPv4Address.DefaultGateway=172.12.0.1 deviceInfo[index].IPv4Address.DhcpEnable=false deviceInfo[index].IPv4Address.IPAddress=172.12.7.102 deviceInfo[index].IPv4Address.SubnetMask=255.255.0.0 deviceInfo[index].IPv6Address.DefaultGateway=2008::1 deviceInfo[index].IPv6Address.DhcpEnable=false deviceInfo[index].IPv6Address.IPAddress=2008::6/112 deviceInfo[index].Mac=00:01:5b:01:44:77 deviceInfo[index].MachineName=YZZ4DZ008D00031 deviceInfo[index].Port=37777 deviceInfo[index].RemoteVideoInputChannels=0 deviceInfo[index].SerialNo=YZZ4DZ008D00031 deviceInfo[index].Vendor=Multi deviceInfo[index].Version=1.200.0.0 deviceInfo[index].VideoInputChannels=1 deviceInfo[index].VideoOutputChannels=16
Comment	Parameters in URL: deviceClass : in range of {VTO, VTH, VTT, VTS, VTNC, SHG} Parameters in Response: index : the array index which starts from 0. Version : Software Version

15.2 Open Platform

15.2.1 Application Start and Stop

- Start Application

Table 15-2

Syntax	http://<server>/cgi-bin/installManager.cgi?action=start&appname=< appname >&appid=< appid >
Method	GET
Description	Start application.
Example	http://192.168.1.250/cgi-bin/installManager.cgi?action=start&appname=FaceDemo&apid=1
Success Return	OK
Comment	Parameters in URL: appname : the application name, appid : optional, the application id

- Stop Application

Table 15-3

Syntax	http://<server>/cgi-bin/installManager.cgi?action=stop&appname=< appname >&appid=< appid >
Method	GET
Description	Stop application.
Example	http://192.168.1.250/cgi-bin/installManager.cgi?action=stop&appname=FaceDemo&apid=1
Success Return	OK
Comment	Parameters in URL: appname : the application name, appid : optional, the application id

15.2.2 Install Application

- Install application with app data

Table 15-4

URL	http://<server>/cgi-bin/dhop.cgi?action=uploadApp
Method	POST
Description	Install application.
[Request Params] (None)	
[Response Params] (OK)	
[Example]	

	POST /cgi-bin/dhop.cgi?action=uploadApp HTTP/1.1 Host: 172.29.2.176 Content-Length: 413124 Content-Type:multipart/form-data;boundary=<boundary>
Request	--<boundary> Content-Disposition: form-data; name="dev_upgrade"; filename="demo.bin" Content-Type: application/octet-stream <app data> --<boundary>--
Response	OK

- Install Application with app download url

Table 15-5

URL	http://<server>/cgi-bin/dhop.cgi?action=installAppByUrl		
Method	GET		
Description	Install application.		
[Request Params] (key=value format in URL)			
Url	string	R	The app download url
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/dhop.cgi?action=installAppByUrl&Url=https://aaa/bbb/ccc/app.p.bin		
Response	OK		

15.2.3 Update Application and License

- Update Application with app download url

Table 15-6

URL	http://<server>/cgi-bin/dhop.cgi?action=updateAppByUrl		
Method	GET		
Description	Update Application		
[Request Params] (key=value format in URL)			
appName	string	R	The app name
Url	string	R	The app download url
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/dhop.cgi?action=updateAppByUrl&appName=xxx&Url=http://aaa/bbb/ccc/app.bin		
Response	OK		

- Update firmware with firmware download url

Table 15-7

URL	http://<server>/cgi-bin/dhop.cgi?action=updateFirmwareByUrl		
Method	GET		
Description	Update firmware		
[Request Params] (key=value format in URL)			
Url	string	R	The firmware download url
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/dhop.cgi?action=updateFirmwareByUrl&Url=https://aaa/bbb/ccc/firmware.bin		
Response	OK		

- Update application license with license download url

Table 15-8

URL	http://<server>/cgi-bin/dhop.cgi?action=updateLicenseByUrl		
Method	GET		
Description	Update application license.		
[Request Params] (key=value format in URL)			
appName	string	R	The app name
Url	string	R	The license download url
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/dhop.cgi?action=updateLicenseByUrl&appName=xxx&Url=https://aaa/bbb/ccc/license.bin		
Response	OK		

- Update application license with license data

Table 15-9

URL	http://<server>/cgi-bin/dhop.cgi?action=uploadLicense		
Method	POST		
Description	Update application license.		
[Request Params] (key=value format in URL)			
appName	string	R	The app name
[Response Params] (OK)			
[Example]			
Request	POST /cgi-bin/dhop.cgi?action=uploadLicense&appName=xxx HTTP/1.1 Host: 172.29.2.176 Content-Type: application/octet-stream Content-Length: <length> <app license data>		
Response	OK		

15.2.4 Uninstall Application

URL	http://<server>/cgi-bin/dhop.cgi?action=uninstall		
Method	GET		
Description	Uninstall application.		
[Request Params] (key=value format in URL)			
appName	string	R	The app name
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/dhop.cgi?action=uninstall&appName=xxx		
Response	OK		

15.2.5 Download Application Log

URL	http://<server>/cgi-bin/dhop.cgi?action=downloadLog		
Method	GET		
Description	Download application log.		
[Request Params] (key=value format in URL)			
appName	string	R	The app name
[Response Params] (log file content)			
[Example]			
Request	http://192.168.1.250/cgi-bin/dhop.cgi?action=downloadLog&appName=xxx		
Response	HTTP/1.1 200 OK Server: Device/1.0 Content-Type: Application/octet-stream Content-Length: <length> <log file data>		

15.3 GPS

15.3.1 Get Capablity

Get the capability of position system.

Syntax	http://<ip>/cgi-bin/GpsControl.cgi?action=getCaps
Method	GET
Description	Get GpsControl caps.
Example	http://192.168.1.250/cgi-bin/GpsControl.cgi?action=getCaps
Success	caps.SUPPORTMODE[0]=""GPS"
Return	caps.SUPPORTMODE[1]=""BEIDOU"

15.3.2 [Config] GPS config

- get config of position system

Syntax	<code>http://<ip>/cgi-bin/configManager.cgi?action=getConfig&name=GPS</code>
Method	GET
Description	Get the config of position system.
Example	<code>http://192.168.1.250/cgi-bin/global.cgi?action= getConfig&name=GPS</code>
Success Return	<pre>Table.GPS[0].Enable[0]=true Table.GPS[1].Enable[1]=false Table.GPS[0].CurMode[0]="GPS" Table.GPS[1].CurMode[1]="BEIDOU" Table.GPS[0].SyncTime[0]=true Table.GPS[1].SyncTime[1]=false</pre>

- set config of position system.

Syntax	<code>http://<ip>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>...]</code>
Comment	In below table, <i>head</i> =GPS
Success Return	OK or ERROR

Appendix:

ParamName	ParamValue type	Description
<i>head</i> .Enable	bool	true/false
<i>head</i> .CurMode	string	"GPS"/"BEIDOU"
<i>head</i> .SyncTime	bool	true/false

15.3.3 Get GPS Status

Syntax	<code>http://<server>/cgi-bin/positionManager.cgi?action=getStatus</code>
Method	GET
Description	Get gps info
Example	<code>http://192.168.1.250/cgi-bin/positionManager.cgi?action=getStatus</code>
Success Return	<pre>status.Time=2011-07-03 21:02:32 status.Longitude=120.175556 status.Latitude=30.186389 status.Speed=30.00 status.Altitude=45.0 status.Bearing=45.3 status.SatelliteCount=11 status.WorkStatus=1</pre>
Comment	<p>Time : time, ex: 2012-01-04 23:30:30.</p> <p>Longitude : Longitude, in degrees, positive East, negative West</p> <p>Latitude : Latitude, in degrees, positive north, negative South</p>

	<p>Speed : speed, unit: km/h Altitude : Altitude, unit: m Bearing : Direction angle , unit: degree SatelliteCount : Satellite number</p> <p>WorkStatus : GPS working status, 0= unlocated, 1= un-differenced positioning, 2= differential positioning, 3= invalid PPS, 6= estimating</p>
--	---

15.4 Lens Function

15.4.1 Get Lens Capability

URL	http://<server>/cgi-bin/LensFunc.cgi?action=getCaps		
Method	GET		
Description	Get Caps		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
Channel	uint	R	Video channel index which starts from 1.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
Caps	object	R	The capacity.
+DepthField	uint8	R	Whether support depth field adjust, 0 : not support, 1 : support
+AngleAdjust	object	O	The angel adjust capacity.
++Support	uint8	R	Whether support angel adjust, 0 : not support, 1 : only support horizontal adjust, 2 : only support vertical adjust, 3 : support both horizontal and vertical adjust.
++MaxVelocity	uint8	R	The max velocity, range from 1 to 8
[Example]			
Request	http://192.168.1.250/cgi-bin/LensFunc.cgi?action=getCaps&Channel=1		
Response	Caps.DepthField=1 Caps.AngleAdjust.Support=3 Caps.AngleAdjust.MaxVelocity=8		

15.4.2 Adjust Angle Continuously

URL	http://<server>/cgi-bin/LensFunc.cgi?action=adjustAngleContinuously		
Method	GET		
Description	Adjust the angel continuously.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
Channel	uint	R	Video channel index which starts from 1.
Direction	uint8	R	The moving direction, 0 : negative X axis direction, 1 : positive X axis direction, 2 : negative Y axis direction, 3 : positive Y axis direction
Velocity	uint8	R	The adjust velocity, range from 1 to MaxVelocity

[Response Params] (OK)	
[Example]	
Request	http://192.168.1.250/cgi-bin/LensFunc.cgi?action=adjustAngleContinuously&Channel=1&Direction=2&Velocity=2
Response	OK

15.4.3 Stop Adjusting Angle

URL	http://<server>/cgi-bin/LensFunc.cgi?action=stopAdjustingAngle		
Method	GET		
Description	Stop adjusting angel		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
Channel	uint	R	Video channel index which starts from 1.
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/LensFunc.cgi?action=stopAdjustingAngle&Channel=1		
Response	OK		

15.4.4 Adjust Depth Field

URL	http://<server>/cgi-bin/LensFunc.cgi?action=adjustDepthField		
Method	GET		
Description	Adjust depth position		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
Channel	uint	R	Video channel index which starts from 1.
DepthField	float	R	The sensor depth position, range from 0 to 1, -1 means reset.
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/LensFunc.cgi?action=adjustDepthField&Channel=1&DepthField=0.3		
Response	OK		

15.4.5 Adjust Depth Field Continuously

URL	http://<server>/cgi-bin/LensFunc.cgi?action=adjustDepthFieldContinuously		
Method	GET		
Description	Adjust depth position continuously.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
Channel	uint	R	Video channel index which starts from 1.
DepthFieldSpee	float	R	The sensor depth position adjust velocity, range from -1 to 1,

d			positive and negative value means difference direction, 0 means stop adjust.
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/LensFunc.cgi?action=adjustDepthFieldContinuously&Channel=1&DepthFieldSpeed=0.03		
Response	OK		

15.4.6 Get Depth Field Status

URL	http://<server>/cgi-bin/LensFunc.cgi?action=getDepthFieldStatus		
Method	GET		
Description	Get depth position status.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
Channel	uint	R	Video channel index which starts from 1.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
Status	object	R	The depth position status.
+Status	string	R	The depth position status, "Normal" : normal status, "AutoDepthField" : adjusting depth position.
+DepthField	float	R	The sensor board relative position, range from 0 to 1
+DepthFieldSteps	uint	R	The depth position total step.
+ResetResult	string	O	The depth position reset result, only valid when reset. "Success" : reset success, "Failed" : reset failed
[Example]			
Request	http://192.168.1.250/cgi-bin/LensFunc.cgi?action=getDepthFieldStatus&Channel=1		
Response	Status.Status=Normal Status.DepthField=0.3 Status.DepthFieldSteps=1000 Status.ResetResult=Success		

15.4.7 Auto Adjust Depth Field

URL	http://<server>/cgi-bin/LensFunc.cgi?action=autoAdjustDepthField		
Method	GET		
Description	Automatic adjust depth position		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
Channel	uint	R	Video channel index which starts from 1.
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.250/cgi-bin/LensFunc.cgi?action=autoAdjustDepthField&Channel=1		
Response	OK		

15.4.8 Scene Correction

Request URL	http://<server>/cgi-bin/api/LensFunc/correctScene			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Channel	int	R	Channel number, starting from 1.	1
Type	enumint	R	Correction type. enumint{ 0: Automatic correction 1: Manual correction }	1
Direction	enumint	O	Rotation direction. This parameter is only valid when the correction method is manual correction. enumint{ 0: Clockwise 1: Counter clockwise }	1
Step	int	O	Speed (range: 0–8). 0 means stop rotation. When the rotation mode is continuous rotation, it means the rotation speed. When the rotation mode is point-based rotation, it means the speed of each rotation. This parameter is only valid when the correction mode is manual correction.	5
Mode	enumint	O	Rotation mode. This parameter is only valid when the correction mode is manual correction. enumint{ 0: Continuous rotation 1: Point-based rotation }	1
Request Example				
<pre>{ "Channel": 1, "Type": 1, "Direction": 1, "Step": 5, "Mode": 1 }</pre>				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				
{}				

15.4.9 Reset Angle

Reset camera lens angle

URL	http://< server >/cgi-bin/LensManager.cgi?action=resetAngle		
Method	GET		
Description	Reset camera lens angle		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
Channel	uint	R	Video channel index which starts from 1.
[Response Params] (OK)			
[Example]			
Request	http://172.27.1.153/cgi-bin/LensManager.cgi?action=resetAngle&Channel=1		
Response	OK		

15.5 FishEye

15.5.1 Get FishEye Capability

The method described in the Section “4.5.12 Get video input capability” should be used first. In the response, it will contain these message "caps.FishEye=false", if the value of the FishEye is true. Then you can use the method described below to get the detail capability.

Table 15-10

Syntax	http://<server>/cgi-bin/devVideoInput.cgi?action=getCapsEx&channel=< ChannelNo >&name=VideoInFishEye
Method	GET
Description	Get fisheye capability.
Example	http://192.168.1.250/cgi-bin/devVideoInput.cgi?action=getCapsEx&channel=1&name=VideoInFishEye
Success Return	caps. Type =Chip caps. MountMode [0]=WallMode caps. MountMode [1]=CeilMode caps. CalibrateMode [0]=Original caps. CalibrateMode [1]=Config caps. CalibrateMode [2]=Panorama caps. EPtzCmd [0]=Up caps. EPtzCmd [1]=Down

Comment	<p>Parameters in Response:</p> <p>Type: string, it can be Chip, Plugin, and ChipAndPlugin. Chip means only support calibrate by device.</p> <p>Plugin means only support calibrate by plugin. ChipAndPlugin means support both.</p> <p>MountMode: string and array.and MountMode means the install mode, it can be WallMode,CeilMode, FloorMode,180CeilMode.</p> <p>CalibrateMode: string and array. It can be Original, Config, Panorama, DoublePanorama, OriginalPlusThreeEPtzRegion, Single, FourEPtzRegion, TwoEPtzRegion, and Normal.</p> <p>EPtzCmd: string and array. It can be ZoomIn, ZoomOut, Up, Down, Left, Right, RotateClock, RotateAntiClock, Stop, TapView, and ShowRegion.</p>
---------	--

15.5.2 [Config] FishEye Setting

- Get FishEye config

Table 15-11

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=FishEye
Method	GET
Description	Get FishEye config.
Example	http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=FishEye
Success Return	head.PlaceHolder =1 head.CalibrateMode =Original
Comment	<p>Parameters in Response:</p> <p>head = table.FishEye[ChannelNo]</p> <p>ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p> <p>PlaceHolder : integer, it can be 1(CeilMode), 2(WallMode), 3(FloorMode).</p> <p>CalibrateMode: string. It can be Original, Config, Panorama, DoublePanorama, OriginalPlusThreeEPtzRegion, Single, FourEPtzRegion, TwoEPtzRegion, and Normal.</p>

- Set FishEye config

Table 15-12

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&< paramName >=< paramValue > [&< paramName >=< paramValue > ...]
Method	GET
Description	Set FishEye config.
Example	<p>Change the placeholder setting of channel 1: http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&FishEye[0].PlaceHolder=1</p> <p>Change the calibratemode setting of channel 1: http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&FishEye[0].Calibrate</p>

	<p>Mode=Panorama</p> <p>Also can use http://192.168.1.250/cgi-bin/configManager.cgi?action=setConfig&FishEye[0].CalibrateMode=Panorama&FishEye[0].PlaceHolder=1 to change the placeholder and calibratemode in one message.</p>
Success Return	OK
Comment	—

15.6 Radar Adaptor

15.6.1 Get Radar Capability

Note: This interface is deprecated. Please use the interface getCapsEx.

Radar equipment detection capability request

Syntax	<a href="http://<server>/cgi-bin/radarAdaptor.cgi?action=getCaps&channel=<ChannelNo>">http://<server>/cgi-bin/radarAdaptor.cgi?action=getCaps&channel=<ChannelNo>
Method	GET
Description	Request radar device detection capabilities
Example	http://192.168.1.250/cgi-bin/radarAdaptor.cgi?action=getCaps&channel=1
Success Return	caps.DetectionRange = 5000 //Zoom in 100 times, while the true value is 50.00 caps.DetectionAngle = 8003 //Zoom in 100 times caps.DetectionHuman = 4000 //Zoom in 100 times
Comment	Parameters in URL: ChannelNo: integer, channel index which starts from 1.

15.6.2 Get Radar Capbility (Enhanced)

Radar equipment caps request

Syntax	<a href="http://<server>/cgi-bin/radarAdaptor.cgi?action=getCapsEx&channel=<ChannelNo>">http://<server>/cgi-bin/radarAdaptor.cgi?action=getCapsEx&channel=<ChannelNo>
Method	GET
Description	Request radar device detection capabilities
Example	http://192.168.1.250/cgi-bin/radarAdaptor.cgi?action=getCapsEx&channel=1
Success Return	caps.DetectionRange = 5000 // Radar detection range caps.DetectionAngle = 8003 // Radar detection angle caps.DetectionHuman = 4000 // Radar human detection range caps.Capacity.Support = true //Power setting enable caps.Capacity.List[0] = 150 // Power allows setting list, no more than 24 choices ... caps.Capacity.List[23] = 300 caps.RadarScene.Support = true // Scene setting enable caps.RadarChannel.Support = true // Radar channel setting enable caps.RadarChannel.List[0] = 1 // Channel allows setting list, no more than 24 choices ... caps.RadarChannel.List[23] = 2

	caps.MovedDetect.Support = true // Device motion detection setting enabled
Comment	Parameters in URL: ChannelNo : integer, channel index which starts from 1.

15.6.3 Get Status

Radar Equipment Status Request

Syntax	http://<server>/cgi-bin/radarAdaptor.cgi?action=getStatus&channel=< ChannelNo >
Method	GET
Description	Request radar device status
Example	http://192.168.1.250/cgi-bin/radarAdaptor.cgi?action=getCaps&channel=1
Success Return	status.State = Normal
Comment	Parameters in URL: ChannelNo : integer, channel index which starts from 1.

Appendix:

ParamName	ParamValue type	Description
State	string	Radar state, the range is { "Normal" , "Abnormal" }

15.6.4 Calculate Real Size

Radar real map size calculation

Syntax	http://<server>/cgi-bin/radarAdaptor.cgi?action=calculateRealSize&channel=< ChannelNo >&distance=< distance >&startPoint[0]=< startX >&startPoint[1]=< startY >&endPoint[0]=< endX >&endPoint[1]< endY >
Method	GET
Description	Calculate radar real map size
Example	http://192.168.0.108/cgi-bin/radarAdaptor.cgi?action=calculateRealSize&channel=1&distance=10.01&startPoint[0]=10&startPoint[1]=20&endPoint[0]=30&endPoint[1]=40
Success Return	realSize.realMap[0]=300 realSize.realMap[1]=400
Comment	Parameters in URL: ChannelNo : integer, channel index which starts from 1. Distance : Segment distance, Unit: m startPoint : Line segment start coordinates endPoint : Line segment end coordinates

Appendix:

ParamName	ParamValue type	Description
realMap	array<integer>	Length and width of actual map, unit:meter

15.6.5 Subscribe Alarm Point Info

Radar track point request

Syntax	http://<server>/cgi-bin/radarAdaptor.cgi?action=attachAlarmPointInfo&channel=<ChannelNo>&[&heartbeat=<Heartbeat>]		
Method	GET		
Description	Request radar track points		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
Channel	int	R	Video channel index which starts from 1
heartbeat	int	O	Send heartbeat interval, range is [1, 60], unit is second. If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client, the heartbeat message are "Heartbeat". If this parameter is not present, its default value is 60.
Example	http://192.168.0.108/cgi-bin/radarAdaptor.cgi?action=attachAlarmPointInfo&channel=1&heartbeat=5		
Success Return	<p>HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<boundary> Connection: closed</p> <p>--<boundary> Content-Type: application/octet-stream Content-Length: <length></p> <p>Channel=1 info[0].PointType=0x80 info[0].RegionNumber=1 info[0].ObjectType=1 info[0].TrackID=1 info[0].Distance=1001 //Zoom in 100 times, while the true value is 10.01 info[0].Angle=2010 //Zoom in 100 times info[0].Speed=501 //Zoom in 100 times</p> <p>..... info[3].PointType=0x80 </p> <p>--<boundary> Content-Type: text/plain Content-Length: 11</p> <p>Heartbeat -<boundary> Content-Type: application/octet-stream Content-Length: <length></p> <p>Channel=1 info[0].PointType=0x80 </p>		

	--<boundary> ...
Comment	<p>Parameters in URL:</p> <p>ChannelNo: integer, channel index which starts from 1.</p> <p>Notice:</p> <p>A detach occurs when the connection is disconnected.</p>

15.6.6 Manual Locate

Radar manual positioning request

Syntax	http://<server>/cgi-bin/radarAdaptor.cgi?action=manualLocate&channel=<ChannelNo>&point[0]=<x>&point[1]=<y>
Method	GET
Description	Request manual radar positioning
Example	http://192.168.1.250/cgi-bin/radarAdaptor.cgi?action=manualLocate&channel=1&point[0]=10&point[1]=10
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>ChannelNo: integer, channel index which starts from 1.</p> <p>point: Pixel coordinates in the code stream</p>

15.6.7 Start Radar Calibration

Radar equipment calibration

Syntax	http://<server>/cgi-bin/radarAdaptor.cgi?action=startRadarCalibration&channel=<ChannelNo>&mode=<mode>&direction=<direction>&step=<step>
Method	GET
Description	Calibration of radar equipment
Example	http://192.168.1.250/cgi-bin/radarAdaptor.cgi?action=startRadarCalibration&channel=1&mode=1&direction=1&step=5
Success Return	OK
Comment	<p>Parameters in URL:</p> <p>ChannelNo: integer, channel index which starts from 1.</p> <p>mode: Calibration mode, 0:auto, 1: manual, 2:init</p> <p>direction: Calibration direction, effective in manual mode , 0 : invalid, 1 : top, 2 : bottom, 3 : left, 4: right, 5: left_top, 6: right_top, 7: left_bottom, 8: right_bottom</p> <p>step: step size</p>

15.6.8 Add Radar Link SD

Radar link SD added

Request URL	http://<server>/cgi-bin/radarAdaptor.cgi?action=addRadarLinkSD
-------------	--

Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Devices	array<object>	R	Add or update the added device login information	
+PassWord	char[32]	R	password, plaintext	"123456"
+UserName	char[128]	R	user name	"admin"
+Port	integer	R	port	5000
+SDLinkIP	char[32]	R	device ip adress	"192.168.1.250"
Channel	integer	R	video channel, which starts from 1.	1
Request Example				
http://192.168.1.250/cgi-bin/radarAdaptor.cgi?action=addRadarLinkSD&Devices[0].SDLinkIP=10.11.17.98&Devices[0].Port=5000&Devices[0].Username=admin&Devices[0].PassWord=123456&Channel=1				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
SD	object	R	dome camera	
+Info	object	R	information	
++ErrorCode	integer	R	errorcode 0 : normal 1 : overtime 2 : Wrong user name or password 3 : The device does not support linkage	0
Response Example				
SD.Info.ErrorCode=0				

15.6.9 Del Radar Link SD

Remove radar link SD

Request URL	http://<server>/cgi-bin/radarAdaptor.cgi?action=delRadarLinkSD			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Devices	array<object>	R	Delete the SD added in radar equipment Empty array will delete all linkage equipment by default	
+SDLinkIP	char[32]	R	device to be deleted	"192.168.1.250"
Channel	integer	R	video channel, which starts from 1.	1
Request Example				
http://192.168.1.250/cgi-bin/radarAdaptor.cgi?action=delRadarLinkSD&Devices[0].SDLinkIP=10.11.17.98&Channel=1				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example

Response Example
OK

15.6.10 Get Link SD State

Linked device status request

Syntax	http://<server>/cgi-bin/radarAdaptor.cgi?action=getLinkSDState&channel=< ChannelNo >&ip[0]=< ip >&ip[1]=< ip > &ip[23]=< ip >
Method	GET
Description	Request linkage device status
Example	http://192.168.0.108/cgi-bin/radarAdaptor.cgi?action=getLinkSDState&channel=1&ip[0]=192.168.1.115&ip[1]=192.168.1.116
Success Return	state[0].SDLinkIP=192.168.1.115 state[0]. State =0 state[0].Channel=1 state[1].SDLinkIP=192.168.1.116 state[1]. State =0 state[1].Channel=2
Comment	Parameters in URL: ChannelNo : integer, channel index which starts from 1. ip : IP address of the linkage device

Appendix:

ParamName	ParamValue type	Description
State	integer	device state, 0: out of time, 1: offline, 2: disconnected

15.6.11 [Config] MapPara

map parameters:

Parameter	Type	R/O	Description	Example
MapPara	object[]	R	One-dimensional array, with one configuration for each video channel	
+RadarCoordinate	object	R	Configuration of radar position on the map	null
++RadarDirectionAngle	double	R	Configuration of radar direction (°)	70.5
++RadarPixelPoint	integer	R	Pixel coordinates of radar on the map	[10, 20]
+MapSize	object	R	Configuration of map size	null
++PixelLine	integer[][]	R	Pixel coordinates of line segment in the image	[[10,10], [20,20]]
++Distance	double	R	The actual length represented by the line segment in the image (m)	10.1

Example

http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=MapPara

Appendix: configuration operation method

configname and **configobject** in the following table correspond to specific configuration names and configuration contents.

special explanation :

In the return of **getConfig**, there is a "table" level above the specific configuration.

In the request of **setConfig**, there is no "table" level above the specific configuration.

- **getConfig**

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name= configName		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
name	string	R	config name
Request Example			
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=MapPara			

Response Params (key=value format in body)
Name
table
+MapPara
Response Example
table.MapPara[0].RadarCoordinate.RadarDirectionAngle=70.5
table.MapPara[0].RadarCoordinate.RadarPixelPoint[0]=10
table.MapPara[0].RadarCoordinate.RadarPixelPoint[1]=20
table.MapPara[0].MapSize.PixelLine[0][0]=10
table.MapPara[0].MapSize.PixelLine[0][1]=10
table.MapPara[0].MapSize.PixelLine[1][0]=20
table.MapPara[0].MapSize.PixelLine[1][1]=20
table.MapPara[0].MapSize.Distance=10.1
table.MapPara[1].RadarCoordinate.RadarDirectionAngle=70.5
...

- **setConfig**

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
MapPara	object[]	R	Specific configuration content
Request Example			
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&MapPara[0].RadarCoordinate.RadarDirectionAngle=70.5&MapPara[0].RadarCoordinate.RadarPixelPoint[0]=10&MapPara[0].RadarCoordinate.RadarPixelPoint[1]=20&MapPara[0].MapSize.PixelLine[0][0]=10&MapPara[0].MapSize.PixelLine[0][1]=10&MapPara[0].MapSize.Distance=10.1			

Response Params (OK in body)
Name
Type
R/O
Description
Example
Response Example
OK

15.6.12 [Config] RadarAnalyseRule

RadarAnalyseRule parameters:

			stop recording when it is stop. Record is not enabled if it is false.	
++VoiceEnable	bool	R	Voice prompt	false
++BeepEnable	bool	R	Buzzer	false
++MessageEnable	bool	R	Upload to the alarm center server.	false
++MailEnable	bool	R	Send email.	false
++TipEnable	bool	R	Local message box prompt	false
++AlarmOutLatch	integer	R	Output delay time (second) after the alarm output stops. Range [10, 300]	10
++AlarmOutEnable	bool	R	Enable alarm output	false
++AlarmOutChannels	int[]	R	List of alarm output channel numbers. One-dimensional array. Each member indicates that the corresponding channel needs to output alarm, and the channel number starts from 0.	[0]
+Enable	bool	R	Enable alarm enabling	false
+AlarmOutNumber	integer	R	Alarm configuration number, which is unique.	1
+SDLinkIP	char[32]	O	Controlled device IP (which is optional and reserved. Ignore this field due to business adjustment).	"192.168.1.250"

Example

<http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=RadarAnalyseRule>

Note: Please refer to appendix in chapter 15.6.11 for configuration operation method

15.6.13 [Config] RadarCalibration

RadarCalibration

Parameter	Type	R/O	Description	Example
RadarCalibration	object[]	R	Configuration of radar and PTZ cameras calibration. One-dimensional array, with one configuration for each video channel.	
+SlopeAngle	double	R	Slope compensation angle.	10.0
+InstallHeight	double	R	Radar installation height (m).	10.5
+CalibrationParas	object[]	R	Calibration parameter.	null
++TiltRecoupAngle	double	R	Vertical compensation angle of linked PTZ camera. The range is -90° to 90°; 0° in the horizontal direction; 0 to 90° below the horizontal direction; -90° to 0° above the	5.0

			horizontal direction.	
++LinkSDHeight	double	R	Installation height of linked PTZ camera (m).	10.5
++CalibrationPos	object[]	R	Array of calibration points.	null
+++Ptz	float[3]	R	PTZ value (normalized value) of calibration points. The first element is the horizontal angle, which is normalized to -1 to 1. The second element is the vertical angle, which is normalized to -1 to 1. The third element is the zoom times, which is normalized to 0 to 1.	[0.95, 0.43, 0.12]
+++AlarmPixelPoint	int[2]	R	Pixel coordinates of calibration points.	[10, 20]
++SDLinkIP	char[32]	R	Controlled device IP.	"192.168.1.250"

Example

<http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=RadarCalibration>

Note: Please refer to appendix in chapter 15.6.11 for configuration operation method

15.6.14 [Config] RadarGuardLine

RadarGuardLine

Parameter	Type	R/O	Description	Example
RadarGuardLine	object[][][10]	R	Two-dimensional array. The first dimension indicates video channel, and the second dimension indicates multiple rule lines (up to 10).	
+AlarmOutNumber	integer	R	Alarm output configuration. Configure the AlarmOutNumber field according to RadarAnalyseRule.	1
+TargetFilter	integer	R	Target filtering bit0: Reserved bit1: Human. bit2: Vehicle. bit3: Animal (Enable by setting it to 1)	0x01
+Polygon	integer	R	Rule point, pixel coordinates [width, height]. Range of number of points [3, 24]	[[0, 0], [128, 128], ..., []]
+Type	char[32]	R	Rule type, alarm or shield type. High Alarm Alarm Shield	"Alarm"
+Enable	bool	R	Whether to enable the rules	true
+RegionNumber	integer	R	Rule line number Range [1, 10]	1
+Name	char[128]	R	Rule name, which cannot be repeated.	"GuardLine1"

Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=RadarGuardLine
```

Note: Please refer to appendix in chapter 15.6.11 for configuration operation method

15.6.15 [Config] RadarLink

RadarLink

Parameter	Type	R/O	Description	Example
RadarLink	object[]	R	One-dimensional array, with one configuration for each video channel	
+RadarLink	bool	R	Enable radar-PTZ linkage	true

Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name= RadarLink
```

Note: Please refer to appendix in chapter 15.6.11 for configuration operation method

15.6.16 [Config] RadarLinkDevice

RadarLinkDevice

Parameter	Type	R/O	Description	Example
RadarLinkDevice	object[][]	R	Remote device linkage configuration. object[][][24] Two-dimensional array. The first dimension indicates video channel, and the second dimension indicates multiple remote linkage devices.	
+DeviceName	char[128]	R	Device name	"XXX North Gate"
+DeviceType	char[32]	R	Device type	"200W PTZ camera"
+Vendor	char[32]	R	Device source	"VVV"
+SDLLinkIP	char[32]	R	Controlled device IP	"192.168.1.250"

Example

```
http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=RadarLinkDevice
```

Note: Please refer to appendix in chapter 15.6.11 for configuration operation method

15.6.17 [Config] RadarPara

RadarPara

Parameter	Type	R/O	Description	Example
RadarPara	object[]	R	Radar function configuration. One-dimensional array, with one configuration for each video channel.	
+Sensitivity	object	O	Recognition sensitivity information.	null
++Level	integer	O	Optional. Skip if this field does not exist. The range is -5 to 5; 0 is the default value; the smaller the value, the higher the human recognition rate; the larger the value, the higher the vehicle recognition rate.	0

+TargetRatio	object	R	Tracking screen ratio for radar-PTZ linkage.	null
++Ratio	integer	R	Reciprocal of screen ratio. Required. Calculate the reciprocal. For example, when the screen ratio is 0.5, Ratio = 2; when the screen ratio is 0.25, Ratio = 4.	1
+RadarChannel	object	R	Radar channel configuration.	null
++Route	integer	R	Channel selection. 1: Channel 1 2: Channel 2	1
+Capacity	object	R	Configuration of radar transmitting power.	null
++Power	integer	R	Radar transmitting power is measured by detection distance (m).	300
+Structured	object	R	Structured information configuration	null
++TrackDisplayTime	integer	O	Range [1, 30] (s).	5
++TrackType	integer	O	0: Disable 1: Trajectory box 2: Trajectory	0
++Enable	bool	R	Enable structured information display	false
+Scene	object	R	Scene configuration	null
++Type	integer	R	Scene selection. 1: Default. 2: Shrub. 3: Spacious. 4: Custom.	1

Example

<http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=RadarPara>

Note: Please refer to appendix in chapter 15.6.11 for configuration operation method

15.6.18 [Config] RadarTrackGlobal

RadarTrackGlobal

Parameter	Type	R/O	Description	Example
RadarTrackGlobal	object[]	R	One-dimensional array, with one configuration for each video channel	
+TrackSwitchMode	char[32]	R	Tracking switching mode Tour/time priority/distance priority Rotation/TimePriority/DistancePriority	"Rotation"
+TrackSwitchTime	integer	R	Tracking switching time (s) Range [1, 15]	5
+TrajectoryTime	integer	R	Trajectory time (s) Range [3, 30]	5
+SectorDisable	bool	R	Enable protection area hiding	false

Example

<http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=RadarTrackGlobal>

Note: Please refer to appendix in chapter 15.6.11 for configuration operation method

15.6.19 [Config] RemoteSDLink

RemoteSDLink

Parameter	Type	R/O	Description	Example
RemoteSDLink	object[]][24]	R	Remote device linkage enabling configuration. Two-dimensional array. The first dimension indicates video channel, and the second dimension indicates multiple remote linkage devices.	
+RadarLink	bool	R	Enable linkage.	true
+SDLinkIP	char[32]	R	Controlled device IP.	"192.168.1.250"
Example				
http://192.168.1.250/cgi-bin/configManager.cgi?action=getConfig&name=RemoteSDLink				

Note: Please refer to appendix in chapter 15.6.11 for configuration operation method

15.7 Water Radar

15.7.1 Acquire Radar Capability

Request URL	http://<server>/cgi-bin/api/WaterRadar/getCaps		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
Request Example			
{}			

Response Params (JSON format in body)	
Name	Type
Support	bool
R/O	
Whether the water conservancy radar function is supported. false: Not Supported true: Supported	
Response Example	
{ "Support": true }	

15.7.2 Get Radar Detection Target Data

Request URL	http://<server>/cgi-bin/api/WaterRadar/getObjectInfo		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
Request Example			
{}			

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
WaterLevel	float	R	Water level, CM	1000.0
Response Example				
{ "WaterLevel": 1000.0 }				

15.8 Water Quality Detection

15.8.1 Get Capability

Acquiring the interface for water quality detection.

Request URL	http://<server>/cgi-bin/api/WaterDataStatServer/getCaps			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Request Example				
{ }				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Support	enumint	O	Supports water quality detection currently. enumint{ 0: No 1: Yes }	0
SupportLocalDataStore	enumint	O	Supports local storage. enumint{ 0: No 1: Yes }	0
Response Example				
{ "Support": 0, "SupportLocalDataStore": 0 }				

15.8.2 Get Real-time Detection Data

The water quality detection data is refreshed every minute, and you can acquire the data in real time.

Request URL	http://<server>/cgi-bin/api/WaterDataStatServer/getWaterData
-------------	--

Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Type	enumchar[32][32]	O	enumchar[32][32]{ "Quality": Water quality category "PH": pH value "NTU": Turbidity value "NH3-N": Ammonia nitrogen value }	["PH", " NTU ", ...]
Request Example				
{ "Type": ["PH", " NTU ", ...] }				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Quality	enumint	O	The smaller the category value, the better the water quality. enumint { 1: I Category 2: II Category 3: III Category 4: IV Category 5: V Category }	1
UploadInfo	object	O	Reported data of water quality detection. Reports all water quality detection data when an alarm is triggered.	
+PH	float	O	pH (1–14)	5.2
+NTU	float	O	Turbidity (0–500 NTU)	200.0
+NH3-N	float	O	Ammonia nitrogen (0–50 mg/L)	20.0
+TN	float	O	Ammonia nitrogen (0–50 mg/L)	25.0
+SD	float	O	Transparency (0–30 m)	20.0
+COD	float	O	Chemical oxygen demand (0–100 mg/L)	50.0
+NN	float	O	Nitrite-nitrogen (0–500 mg/L)	20.0
+DO	float	O	Dissolved oxygen (0–10 mg/L)	5.0
+Chl-a	float	O	Chlorophyll A (0–300 ug/L)	200.0
+TP	float	O	Total phosphorus (0–5 mg/L)	2.5
+CODMn	float	O	Permanganate index (0–100 mg/L)	20.0
+SS	float	O	Floating objects (0–1000 mg/L)	20.0
+BOD5	float	O	Biochemical oxygen demand for 5 days (0–50 mg/L)	25.0
+NO3-N	float	O	Nitrate (0–500 mg/L)	20.0
+TSI	float	O	Eutrophic index (no range)	20.0
+BlackSmellyLevel	enumchar[32]	O	Black and odorous level enumchar[32]{	"Light"

			"Normal" "Light" "Heavy" }	
FlunkType	enumchar[32][32]	O	Parameter value exceeds its threshold enumchar[32][32]{ "PH": pH value "NTU": Turbidity value "NH3-N": Ammonia nitrogen value }	["PH", "NTU", ...]

Response Example

```
{
  "Quality": 1,
  "UploadInfo": {
    "PH": 5.2,
    "NTU": 200.0,
    "NH3-N": 20.0,
    "TN": 25.0,
    "SD": 20.0,
    "COD": 50.0,
    "NN": 20.0,
    "DO": 5.0,
    "Chl-a": 200.0,
    "TP": 2.5,
    "CODMn": 20.0,
    "SS": 20.0,
    "BOD5": 25.0,
    "NO3-N": 20.0,
    "TSI": 20.0,
    "BlackSmellyLevel": "Light"
  },
  "FlunkType": ["PH", "NTU", ...]
}
```

15.8.3 Start Find Water Quality Report

Searching for water quality report.

Request URL	http://<server>/cgi-bin/api/WaterDataStatServer/startFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
condition	Object	O	Search conditions of water quality detection	
+StartTime	char[20]	O	Start time (temporarily accurate to the hour)	"5/25/2010 00:00:00"
+Type	enumchar[32][32]	O	enumchar[32][32]{ "Quality": Water quality category	["PH", "NTU"]

			"PH": pH value "NTU": Turbidity value "NH3-N": Ammonia nitrogen value }	
+PresetID	int[]	O	Preset ((one-dimensional array))	[1]
+EndTime	char[20]	O	End time (temporarily accurate to the hour)	"2010-05-25 00:00:00"

Request Example

```
{
  "condition": {
    "StartTime": "2010-05-25 00:00:00",
    "Type": ["PH", " NTU"],
    "PresetID": [1],
    "EndTime": "2010-05-25 00:00:00"
  }
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
token	uint32	O	Obtained search token	2342343
totalCount	uint32	O	Total number of qualified results	233

Response Example

```
{
  "token": 2342343,
  "totalCount": 233
}
```

15.8.4 Get Find Water QUality Report Search Results

Searching for water quality report.

Request URL	http://<server>/cgi-bin/api/WaterDataStatServer/doFind		
Method	POST		

Request Params (JSON format in body)

Name	Type	R/O	Description	Example
token	uint32	O	Search token	46878
beginNumber	uint32	O	Starting sequence number of the search. It means that the search starts from the beginNumber records, and returns count records. $0 \leq \text{beginNumber} \leq \text{totalCount}-1$.	0
count	int32	O	Number of traffic statistics for each search	24

Request Example

```
{
  "token": 46878,
  "beginNumber": 0,
  "count": 24
}
```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
found	uint32	O	Number of searched entries	12
info	object[]	O	Traffic statistics. It is an array and each element represents a traffic record that satisfies the condition.	
+StartTime	char[20]	O	Start time	2018-03-03 00:00:00",
+Quality	enumint	O	enumint { 1: I Category 2: II Category 3: III Category 4: IV Category 5: V Category }	1
+PH	float	O	pH (1–14)	5.5
+NTU	float	O	Turbidity (0–500 NTU)	250.0
+NH3-N	float	O	Ammonia nitrogen (0–50 mg/L)	25.0
+TN	float	O	Total nitrogen (0–50 mg/L)	25.0
+SD	float	O	Transparency (0–30 m)	20.0
+COD	float	O	Chemical oxygen demand (0–100 mg/ L)	20.0
+NN	float	O	Nitrite-nitrogen (0–500 mg/ L)	250.0
+Chl-a	float	O	Chlorophyll A (0–300 ug/ L)	20.0
+TP	float	O	Total phosphorus (0–5 mg/L)	2.5
+CODMn	float	O	Permanganate index (0–100 mg/L)	20.0
+SS	float	O	Floating objects (0–1000 mg/ L)	500.0
+BOD5	float	O	Biochemical oxygen demand for 5 days (0–50 mg/L)	25.0
+NO3-N	float	O	Nitrate (0–500 mg/ L)	250.0
+TSI	float	O	Eutrophic index (no range)	25.0
+BlackSmellyLevel	enumchar[32]	O	Black and odorous level enumchar[32]{ "Normal" "Light" "Heavy" }	"Light"
+DO	float	O	Dissolved oxygen (0–10 mg/L)	20.0

Response Example

```
{
  "found": 12,
  "info": [
    {
      "StartTime": "2012-03-14 00:00:00",
      "Quality": 1,
      "PH": 5.5,
      "NTU": 250.0,
      "NH3-N": 25.0,
      "TN": 25.0,
      ...
    }
  ]
}
```

```

        "SD": 20.0,
        "COD": 20.0,
        "NN": 250.0,
        "Chl-a": 20.0,
        "TP": 2.5,
        "CODMn": 20.0,
        "SS": 500.0,
        "BOD5": 25.0,
        "NO3-N": 250.0,
        "TSI": 25.0,
        "BlackSmellyLevel": "Light",
        "DO": 20.0
    },...{}]
}

```

15.8.5 Stop Find Water Quality Report

Stopping the search of water quality detection database.

Request URL	http://<server>/cgi-bin/api/WaterDataStatServer/stopFind		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
token	uint32	O	Search token
Request Example			
{	"token": 46284		
}			

Response Params (JSON format in body)
Name
Type
R/O
Description
Example
Response Example
{}

15.9 Advertisement

15.9.1 Changing Stay Time of Advertisement Image

Change the stay time of advertisement image.

Syntax	http://<server>/cgi-bin/VideoOutput.cgi?action=changeSustain&Sustain=5		
Method	GET		
Parameter Format	key=value format at URL		
Parameter	Type	R/O	Description
Sustain	int	R	Stay time of each Image.
Example	5		
Complete Example	http://<server>/cgi-bin/VideoOutput.cgi?action=changeSustain&Sustain=5		

15.9.2 Acquiring Video and Image Files Played by the Device

Acquire video and image files played by the device.

Syntax	http://<server>/cgi-bin/VideoOutput.cgi?action=queryDeliveredFile			
Method	GET			
Parameter Format	key=value format at URL			
Parameter	Type	R/O	Description	Example
Request Example				
http://<server>/cgi-bin/VideoOutput.cgi?action=queryDeliveredFile				

Parameter Format	key=value format at URL			
Parameter	Type	R/O	Description	Example
params	object[]	R	All play information	
+Enable	bool	O	Enable play function.	true
+Number	int	R	The current advertisement plan number. The caller can use this number to set up different advertisement plans.	0
+TimeSection	TimeSchedule	R	Advertisement playing period. It is a two-dimensional array in which the first seven elements correspond to seven days of a week, and the eighth element corresponds to holidays, with a maximum of six periods per day. The eighth element can be left blank or entered as null, indicating that the holiday period is not supported.	[["1 00:00:00-07:00:00", "2 09:00:00-17:30:00", "3 17:30:00-23:59:59"], ..., []]
+Name	char[128]	O	Advertisement name	"Happily holliday"
+StartTime:	char[20]	O	Start time	"2016-08-10 10:00:00"
+EndTime:	char[20]	O	End time	"2016-08-10 12:00:00"
+Mode	char[16]	O	Video play mode. Once: Each file in the list is played once. Repeat: Files in the list are played repeatedly until end time. Alone: Cut-in play (playing alone).	"Once"
+FileList	object[]	O	Video file list.	Up to 20 file lists are supported.
++Downloaded	bool	O	Check whether the file has been downloaded to the device.	true
++FileType	enumchar[16]	O	File type. enumchar[16]{ "Video": Video file }	"Video"

			<pre>"Image": Image file "Audio": Audio file }</pre>	
++LocalPath	char[128]	O	The path where the file is downloaded to the device.	"/mnt/appdata/Publish/1.dav"
++Sustain	int	O	Stay time of each Image, which is valid only when FileType is Image. Unit: Second.	5
++URL	char[512]	O	The resource address of the file, and the maximum length is 512 bytes. It has the same length as the VideoOutput.deliveryFile interface (used by video intercom and delivered at the same time).	"ftp://192.168.1.250/1 дав"
++TimeSection	TimeSchedule	O	Advertisement playing period. It is a two-dimensional array in which the first seven elements correspond to seven days of a week, and the eighth element corresponds to holidays, with a maximum of six periods per day. The eighth element can be left blank or entered as null, indicating that the holiday period is not supported.	[["1 00:00:00-07:00:00", "1 09:00:00-17:30:00", "1 17:30:00-23:59:59"], ..., []]
++Size	int	O	Files size. Unit: Byte.	102400
++ID	int	O	File number.	3

Reponse Example

```

params[0].Enable=true
params[0].Number=0
params[0].TimeSection[0][0]=6 00:00:00-23:59:59
params[0].TimeSection[0][1]=0 00:00:00-23:59:59
params[0].TimeSection[0][2]=0 00:00:00-23:59:59
params[0].TimeSection[0][3]=0 00:00:00-23:59:59
params[0].TimeSection[0][4]=0 00:00:00-23:59:59
params[0].TimeSection[0][5]=0 00:00:00-23:59:59
params[0].TimeSection[1][0]=6 00:00:00-23:59:59
params[0].TimeSection[1][1]=0 00:00:00-23:59:59
params[0].TimeSection[1][2]=0 00:00:00-23:59:59
params[0].TimeSection[1][3]=0 00:00:00-23:59:59
params[0].TimeSection[1][4]=0 00:00:00-23:59:59
params[0].TimeSection[1][5]=0 00:00:00-23:59:59
...
params[0].Name="Happy"
params[0].StartTime="2016-08-10 10:00:00"
params[0].EndTime="2016-08-10 12:00:00"
params[0].Mode="Once"
params[0].FileList.Downloaded=true
params[0].FileList.FileType="Video"

```

```
params[0].FileList.LocalPath="/mnt/appdata/Publish/1.dav"
params[0].FileList.Sustain=5
params[0].FileList.URL="ftp://192.168.1.108/1.dav"
params[0].FileList.Size=102400
params[0].FileList.ID=3
...

```